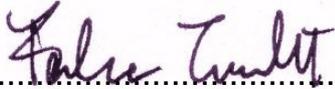
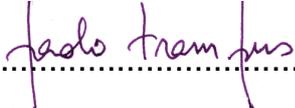




AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 1/131
Filename: *ASTEDetailedDesign6*

TITLE	ASTE DETAILED DESIGN
DOCUMENT TYPE	REPORT
DOC No.	AMST/ STED /1/A
ISSUE No.	6
DATE	Oct 06

Prepared by:	F. TURCHET	(CARSO)	
Approved by:	P. TRAMPUS	(CARSO)	



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
 ISSUE: 6
 DATE: Oct 06
 PAGE: 2/131
 Filename: ASTEDetailedDesign6

CHANGE RECORD

Issue	Date	Affected Pages	Description of Change	Change Authority
2	08/11/05	6	Added "Power plane"	F. T.
2	08/11/05	15-25	Updated schematics	F.T.
3	21/11/05	12	Updated bill of materials	F.T.
3	21/11/05	4,15-24	Updated schematics	F.T.
4	28/11/05	20,23,24	Updated schematics	F.T.
5	03/08/06	12,15-25	Updated schematics	F.T.
6	Oct 06		AddedVHDL code	

Table of Contents

1.	CIRCUIT DESCRIPTION	5
1.1.	RAM PROGRAM	5
1.2.	RAM DATA	5
1.3.	CONNECTORS.....	5
1.4.	DSP.....	5
1.5.	POWER SUPPLY CAMERAS	5
1.6.	RAM IMAGE BUFFERS.....	6
1.7.	CAMERA INTERFACE	6
1.8.	DSP POWER SUPPLY	6
1.9.	LATCHUP CONTROL	6
1.10.	ADC CONVERTERS.....	7
1.11.	POWER PLANE.....	7
2.	ASTE MEMORY MAP.....	9
3.	CONNECTORS.....	10
3.1.	CONNECTORS P1 - CAMERA 0 AND P2 - CAMERA 1	10
3.2.	CONNECTOR P3 – BOARD TO BOARD CONNECTOR	11
3.3.	VME CONNECTOR J1.....	12
3.4.	VME CONNECTOR J2.....	13
4.	BILL OF MATERIALS.....	14
5.	VHDL	17
5.1.	Serial.....	18
5.1.1.	top_level.vhd	18
5.1.2.	dsp_data.vhd	37
5.1.3.	mux_add.vhd	38
5.1.4.	mux_data.vhd.....	39
5.1.5.	counter_addr.vhd	40
5.1.6.	serial.vhd.....	41
5.1.7.	uart_top.vhd	42
5.1.8.	intface.vhd	46
5.1.9.	rxcvr.vhd	52
5.1.10.	txmitt.vhd.....	57
5.1.11.	counter.vhd	62
5.1.12.	timer_data.vhd	62
5.1.13.	hk.vhd	64
5.1.14.	hk_adc.vhd.....	67
5.1.15.	reg_sram.vhd	68
5.1.16.	bus_cont.vhd.....	69
5.1.17.	timeout.vhd	72
5.2.	Receiver.....	74
5.2.1.	top_level.vhd	75
5.2.2.	div16.vhd	84
5.2.3.	div2500.vhd	85
5.2.4.	rx_complete.vhd	86
5.2.5.	rx_read.vhd	89
5.2.6.	rx_ck_rec.vhd	90



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 4/131
Filename: *ASTEDetailedDesign6*

5.2.7.	my_and7.vhd	92
5.2.8.	my_and8.vhd	92
5.2.9.	rx_sync_gen.vhd	93
5.2.10.	rx_data_rec.vhd.....	94
5.2.11.	my_xor14.vhd.....	97
5.2.12.	rx_ctrl_dec.vhd.....	98
5.2.13.	rx_err_ctrl_machine.vhd.....	99
5.2.14.	rx_parity_counter.vhd.....	102
5.2.15.	rx_timeout_ctrl.vhd.....	104
5.2.16.	rx_data_mux.vhd	105
5.2.17.	rx_mux.vhd.....	106
5.2.18.	tx_count.vhd	108
5.2.19.	time_tag_counter.vhd.....	109
5.2.20.	tx_complete.vhd.....	113
5.2.21.	tx_data_gen.vhd.....	115
5.2.22.	my_xor14.vhd.....	116
5.2.23.	tx_ds_gen.vhd.....	117
6.	SCHEMATICS.....	120

List of Tables

1. CIRCUIT DESCRIPTION

1.1. RAM PROGRAM

In the first schematics sheet the program's FLASH and RAM are reported. After boot, the program is copied from the FLASH to the RAM memory, in order to increase execution speed. The buffers convert 3.3V to 5V. FLASH is 4Mword x 48 bit large and the first 32kword (boot memory of ADSP21020) are write protected when jumper J1 is open. The total program RAM is 256kword x 48 bit.

Chip select:

- PMS0 for FLASH selection
- PMS1 for RAM selection

The resistive networks delay addresses and PMSx to keep them stable until the end of a memory write process (see ADSP21020 manual).

1.2. RAM DATA

The data RAM is composed of two pages:

- Page 0: chip select DMS0, 256kword x 40 bit.
- Page 1: chip select DMS1, 256kword x 16 bit. The upper 16 bits are keep low in a reading process by the buffer U25 controlled by 16BIT signal.

Resistors networks are used for delays as in memory program.

1.3. CONNECTORS

VME bus connectors: only some power supply pins are compatible with VME standard connector.

U51-U54 Dallas temperature sensors.

U55 LVDS signal receiver for synchronism (SYNC).

U50 serial interfaces buffer.

1.4. DSP

ADSP 21020

Pullup resistors on CLKIN keep Vhigh over 3.0V (see ADSP21020 datasheet).

1.5. POWER SUPPLY CAMERAS

Cameras power supply:

+12Cx: connected to +12B of VME by transistor Q302 used as switch and max current limiter. The other parts of the circuit measure the current (using the ADC in another sheet) and control the current limiter (typical 220mA).

+5Cx: connected to +5B of VME by U321 used as latchup protection (28mA).

+3.3Cx: derived from +5B with latchup protection (260mA).

The total +5B current used is measured by ADC.

-8Cx: derived from -12B with transistor and zener diode. R353-R356 are used as current limitation. The current is measured by ADC (45mA).

1.6. RAM IMAGE BUFFERS

There are two image memories of 256kword x 16bit and the FPGA connects them to the different buses.

Depending on FLAG2 memory 0 is connected with the camera receiver (camera interface) and memory 1 to ADSP21020 (chip select DMS2), or vice versa. When the camera interface is writing to a memory an image, the ADSP21020 can read the other memory buffer.

Using chip select DMS3 and some addresses, the ADSP21020 can select the following functions, present inside the FPGA:

- Milliseconds time counter (32?? bit) cleared from SYNC, readable in two steps: high and low 16 bit.
- 2 serial interfaces (?? Baud) for external communication.
- control of signal 16BIT when DMS1, DMS2, DMS3 are selected for reading (DMRD).
- Camera interfaces for image and commands.
- ADC's serial interfaces.

1.7. CAMERA INTERFACE

The interface is composed by a FPGA with the cameras receivers and the LVDS buffers.

Quartz Q2 is used only if Q1 frequency is not a 20MHz multiple.

J6 controls cameras default settings, J7 is for debug.

Buffers U35 and U36 act as interface with the second ADSP21020 board connected via P3.

1.8. DSP POWER SUPPLY

There are the DSP and memories power supplies with a measure of the total current and four latchup protections: +5V DSP, +3.3V program memory, +3.3V data memory and +2.5V FPGAs internal.

The LVDS buffers 3.3V power supply with current measure has a current limitation made by R150 and R151.

1.9. LATCHUP CONTROL

Circuits for switching off in case of latchup.

Camera 0 and 1 are turned on by setting FLAG0 and FLAG1. If a latchup occurs they turn off and remain off until they are turned on again by FLAGx. These circuits are insensitive to the FAULT signal during the turn on of the cameras.

The RESET of DSP is kept low during the power-up (EVDD). In case of latchup Q204 pulls down EVDD and after

some ms the circuit turns on the system again. DACx_ON/OFF_DSP are the signals L10_IN_H and L10_IN_C from the backplane and they choose which board to turn on: if they are both at 0V is on the board 1, in the other cases is on the board 0. The transistors act as a trigger insensible to noise.

1.10. ADC CONVERTERS

There are the ADC, the ADC power supply and the onboard temperature sensors (NTC thermistors). ADC0 measures the cameras currents and buffer U111 connects it to the board 1.

1.11. POWER PLANE

In the next picture is visible the power plane divided in three zones:

- Orange: VCC* (+3.3 for program memory)
- Red: EVDD (+5V for ADSP21020)
- Yellow: VCC (+3.3 for data memory and FPGA)

AMICA FOR AMS ASTE Detailed Design



2. ASTE MEMORY MAP

PROGRAM MEMORY	Type	Start	End	Length	Size (bits)	Memory select	Wait states	Access
Program flash	flash	0x000000	0x3FFFFFF	4M	48	PMS0	2?	R/(W)
Program ram	ram	0x400000	0x43FFFF	256K	48	PMS1	0	R/W

DATA MEMORY								
Data ram	ram	0x000000	0x03FFFF	256K	40	DMS0	0	R/W
Data ram	ram	0x040000	0x07FFFF	256K	16	DMS1	0	R/W
Image buffer	ram (FPGA)	0x080000	0xBFFFF	256K	16	DMS2	0?	R
serial port 1	FPGA	0xC0000	0xC0003	4	8	DMS3	0	R/W
serial port 2	FPGA	0xC0004	0xC0007	4	8	DMS3	0	R/W
FPGA register	FPGA	0xC0008	0xC0008	1	??	DMS3	0	R/W
Timer counter	FPGA	0xC0009	0xC0009	1	32	DMS3	0	R
camera register	FPGA	0xC000A	0xC000A	1	16	DMS3	0	W

3. CONNECTORS

3.1. CONNECTORS P1 - CAMERA 0 AND P2 - CAMERA 1

<i>Pin</i>	<i>Signal name</i>	<i>Description</i>
1	NC	
2	-8_Cx	-8V power supply
3	+3.3_Cx	3.3V power supply
4	GND	
5	+5_Cx	5V power supply
6	+12_Cx	12V power supply
7	GND	
8	GND	
9	Cx_Din+	LVDS data
10	Cx_Sin+	LVDS data
11	Cx_Sout-	LVDS data
12	Cx_Dout-	LVDS data
13	NC	
14	-8_Cx	-8V power supply
15	GND	
16	+3.3_Cx	3.3V power supply
17	GND	
18	+5_Cx	5V power supply
19	+12_Cx	12V power supply
20	GND	
21	Cx_Din-	LVDS data
22	Cx_Sin-	LVDS data
23	GND	
24	Cx_Sout+	LVDS data
25	Cx_Dout+	LVDS data

3.2. CONNECTOR P3 – BOARD TO BOARD CONNECTOR

<i>Pin</i>	<i>Signal name</i>	<i>Description</i>
1	NC	
2	OUT_2	Cameras data
3	OUT_4	Cameras data
4	A0_2	Cameras data
5	A0_4	Cameras data
6	GND	
7	CONN_ADC_SCLK	ADC0 signal
8	CONN_ADC_DIN	ADC0 signal
9	CONN.EXT_F1	On/off for camera 1
10	GND	
11	GND	
12	NC	
13	NC	
14	OUT_1	Cameras data
15	OUT_3	Cameras data
16	GND	
17	A0_1	Cameras data
18	A0_3	Cameras data
19	CONN_ADC_CS	ADC0 signal
20	CONN_ADC_DOUT	ADC0 signal
21	GND	
22	CONN.EXT_F0	On/off for camera 0
23	CONN.EXT_ON/OFF_DSP	On/off for DSP board 1
24	GND	
25	NC	



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
 ISSUE: 6
 DATE: Oct 06
 PAGE: 12/131
 Filename: *ASTEDetailedDesign6*

3.3. VME CONNECTOR J1

<i>Row A</i>	<i>Signal</i>	<i>Row B</i>	<i>Signal</i>	<i>Row C</i>	<i>Signal</i>
1		1		1	
2		2		2	
3		3		3	
4		4		4	
5		5		5	
6		6		6	
7		7		7	
8		8		8	
9	GND	9		9	GND
10		10		10	
11	GND	11		11	
12		12		12	
13		13		13	
14		14		14	
15	GND	15		15	
16		16		16	GND_DS6H
17		17		17	DSIO6HOT
18		18		18	DSCH6HOT
19	GND	19		19	GND_DS6C
20		20	GND	20	DSIO6CLD
21		21		21	DSCH6CLD
22		22		22	
23		23	GND	23	
24		24		24	
25		25		25	
26		26		26	
27		27		27	
28		28		28	
29		29		29	
30		30		30	
31	-12B	31		31	+12B
32	+5B	32	+5B	32	+5B

3.4. VME CONNECTOR J2

<i>Row A</i>	<i>Signal</i>	<i>Row B</i>	<i>Signal</i>	<i>Row C</i>	<i>Signal</i>
1		1	+5B	1	AST A0+
2		2	GND	2	AST A0-
3		3		3	
4	+12B	4		4	+12B
5		5		5	
6	-12B	6		6	-12B
7		7		7	
8		8		8	DAC0 ON/OFF DSP
9		9		9	DAC1 ON/OFF DSP
10		10		10	
11		11		11	
12	GND	12	GND	12	GND
13	+5B	13	+5B	13	
14		14		14	
15		15		15	
16		16		16	
17		17		17	
18		18		18	
19		19		19	
20		20		20	
21		21		21	
22	GND	22	GND	22	GND
23		23		23	
24		24		24	AST TXD1
25		25		25	AST RXD1
26		26		26	AST TXD0
27		27		27	AST RXD0
28		28		28	
29		29		29	
30		30		30	
31		31	GND	31	
32		32	+5B	32	



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
 ISSUE: 6
 DATE: Oct 06
 PAGE: 14/131
 Filename: ASTEDetailedDesign6

4. BILL OF MATERIALS

Item	Quantity	Reference	Part
1	1	C1	100pF
2	132	C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39, C40, C42, C43, C44, C46, C47, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C101, C103, C104, C105, C107, C109, C111, C112, C114, C116, C117, C119, C121, C123, C125, C127, C128, C130, C132, C133, C134, C135, C136, C141, C142, C143, C144, C151, C153, C154, C170, C171, C172, C173, C174, C175, C177, C178, C179, C180, C181, C182, C200, C205, C206, C241, C260, C300, C302, C304, C305, C306, C308, C320, C321, C323, C324, C328, C330, C331, C350, C352, C353, C354, C355, C400, C402, C403, C405, C420, C422, C426, C428, C429, C450, C451, C452, C453	100nF
3	40	R14, C45, C48, C61, C62, C64, C100, C102, C106, U107, R122, C122, R200, Q200, R201, C201, C202, C204, C243, C262, C301, C303, C307, C309, U321, C322, C325, C326, R330, C332, C351, C401, C404, C406, U420, C421, C423, C424, R427, C430	NU
4	19	C59, C60, C63, C65, C66, C108, C110, C113, C115, C118, C120, C150, C152, C176, C183, C327, C329, C425, C427	22uF
5	5	C124, C126, C129, C356, C456	6.8uF
6	3	C131, C140, C145	100nF NU
7	3	C203, C242, C261	10nF
8	4	D140, D141, D142, D143	BAS16 NU
9	5	D150, D151, D200, D240, D260	BAS16
10	2	D450, D350	BZX84 8V2
11	1	JP1	WRITE PROTECT
12	2	J2, J1	VME96P
13	1	J3	CON12AP



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
 ISSUE: 6
 DATE: Oct 06
 PAGE: 15/131
 Filename: ASTEDetailedDesign6

14	1	J4	CON24
15	1	J5	CON10
16	2	J6, J7	CON16AP
17	1	J8	CON10AP
18	5	M1, M2, M3, M4, M5	mthole4
19	3	M6, M7, M8	mthole1
20	6	M9, M10, M11, M12, M13, M14	mthole2
21	5	NTC1, NTC2, NTC3, NTC4, NTC5	5k
22	2	P2, P1	CONNECTOR DB25
23	1	P3	CONNECTOR DB25_NU
24	1	Q1	40MHZ_NU
25	1	Q2	20MHZ
26	13	Q201, Q204, Q205, Q206, Q207, Q208, Q209, Q240, Q243, Q260, Q263, Q301, Q401	BC817-40L
27	12	Q202, Q203, Q241, Q242, Q261, Q262, Q300, Q350, Q351, Q400, Q450, Q451	BC807-40L
28	2	Q302, Q402	IRFR5305
29	7	R3, R4, R6, R7, R8, R11, R12	100x8
30	20	R5, R106, R154, R203, R204, R213, R214, R218, R227, R230, R304, R308, R310, R324, R328, R335, R405, R407, R425, R432	10k
31	8	R10, R102, R252, R272, R302, R322, R402, R422	1.8k
32	10	R16, R17, R18, R104, R220, R222, R233, R326, R361, R459	1k
33	37	R50, R51, R103, R140, R141, R152, R207, R208, R209, R219, R223, R224, R241, R242, R243, R244, R248, R250, R261, R262, R263, R264, R268, R270, R303, R305, R307, R309, R312, R314, R325, R403, R404, R406, R409, R411, R423	47k
34	2	R101, R100	0.22
35	26	R105, R132, R133, R134, R135, R136, R153, R205B, R206, R212, R215, R216, R240, R246, R247, R260, R266, R267, R323, R327, R350, R358, R360, R424, R450, R458	22k
36	6	R107, R112, R117, R306, R352, R452	2.7k
37	8	R108, R109, R110, R111, R113, R114, R115, R116	10
38	12	R118, R119, R120, R121, R210, R211, R228, R229, R231, R232, R251, R271	100
39	9	R123, R353, R354, R355, R356, R453, R454, R455, R456	150
40	11	R124, R127, R205, R225, R226, R249, R269, R313, R351, R410, R451	100k



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
 ISSUE: 6
 DATE: Oct 06
 PAGE: 16/131
 Filename: *ASTEDetailedDesign6*

41	2	R128,R129	100k NU
42	1	R142	47k NU
43	1	R143	47k x8
44	2	R150,R151	33
45	1	R202	4.7k
46	1	R217	470k
47	1	R221	10k/47k
48	7	R245,R265,R311,R357,R359, R408,R457	68k
49	8	R300,R301,R320,R321,R400, R401,R420,R421	0.47
50	8	R331,R332,R333,R334,R428, R429,R430,R431	22
51	16	U1,U2,U3,U10,U11,U12,U13, U14,U21,U22,U23,U24,U25, U26,U27,U28	74LVT16244
52	9	U4,U5,U6,U17,U18,U19,U20, U29,U30	K6R4016
53	3	U7,U8,U9	AM29LV641
54	1	U16	ADSP-21020
55	2	U31,U32	A54SX32A
56	2	U55,U33	SN65LVDT390
57	1	U34	SN65LVDS391
58	3	U35,U36,U111	74LVT16244 P NU
59	1	U50	74LVT16244 P
60	4	U51,U52,U53,U54	DS18S20Z
61	3	U100,U300,U320	LM6142
62	2	U101,U105	MAX891L
63	6	U102,U104,U108,U150,U322, U421	MAX1792 3, 3
64	3	U103,U323,U422	MAX892L
65	1	U106	MAX1792 2, 5
66	2	U109,U110	MAX1281



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 17/131
Filename: *ASTEDetailedDesign6*

5. VHDL

5.1. Serial

5.1.1. top_level.vhd

```
-- top_level.vhd

library ieee;
use ieee.std_logic_1164.all;
library a54sxa;
use a54sxa.all;

entity top_level is
    port ( stx_p           : out      std_logic;
          strb_p          : out      std_logic;

          ck20_p          : in       std_logic;
          rst_cpu_p       : out      std_logic;
          rst0_p          : in       std_logic;
          sync_p          : in       std_logic;
          addr_dsp0_p     : in       std_logic;
          addr_dsp1_p     : in       std_logic;
          addr_dsp2_p     : in       std_logic;
          addr_dsp3_p     : in       std_logic;
          addr_dsp4_p     : in       std_logic;
          addr_dsp5_p     : in       std_logic;
          addr_dsp6_p     : in       std_logic;
          addr_dsp7_p     : in       std_logic;
          addr_dsp8_p     : in       std_logic;
          addr_dsp9_p     : in       std_logic;
          addr_dsp10_p    : in       std_logic;
          addr_dsp11_p    : in       std_logic;
          addr_dsp12_p    : in       std_logic;
          addr_dsp13_p    : in       std_logic;
          addr_dsp14_p    : in       std_logic;
          addr_dsp15_p    : in       std_logic;
          addr_dsp16_p    : in       std_logic;
          addr_dsp17_p    : in       std_logic;

          data_dsp8_p     : inout     std_logic;
          data_dsp9_p     : inout     std_logic;
          data_dsp10_p    : inout     std_logic;
          data_dsp11_p    : inout     std_logic;
          data_dsp12_p    : inout     std_logic;
          data_dsp13_p    : inout     std_logic;
          data_dsp14_p    : inout     std_logic;
          data_dsp15_p    : inout     std_logic;
          data_dsp16_p    : inout     std_logic;
          data_dsp17_p    : inout     std_logic;
          data_dsp18_p    : inout     std_logic;
          data_dsp19_p    : inout     std_logic;
          data_dsp20_p    : inout     std_logic;
          data_dsp21_p    : inout     std_logic;
          data_dsp22_p    : inout     std_logic;
          data_dsp23_p    : inout     std_logic;
```

```
dms1_p      : in      std_logic;
dms2_p      : in      std_logic;
dms3_p      : in      std_logic;
dmrd_p      : in      std_logic;
dmwr_p      : in      std_logic;
flag2_p     : in      std_logic;
flag3_p     : in      std_logic;

bit16_p     : out     std_logic;
irq0_p      : out     std_logic;
irq1_p      : out     std_logic;
irq3_p      : out     std_logic;

data_cam0_p : inout   std_logic;
data_cam1_p : inout   std_logic;
data_cam2_p : inout   std_logic;
data_cam3_p : inout   std_logic;
data_cam4_p : inout   std_logic;
data_cam5_p : inout   std_logic;
data_cam6_p : inout   std_logic;
data_cam7_p : inout   std_logic;
data_cam8_p : inout   std_logic;
data_cam9_p : inout   std_logic;
data_cam10_p : inout  std_logic;
data_cam11_p : inout  std_logic;
data_cam12_p : inout  std_logic;
data_cam13_p : inout  std_logic;
data_cam14_p : inout  std_logic;
data_cam15_p : inout  std_logic;

cam_we_p    : out     std_logic;
sel_cam_p   : out     std_logic;

rst_row_p   : in      std_logic;
end_im_p    : in      std_logic;
im_we_p     : in      std_logic;
dir_p       : out     std_logic;

addr1_ram0_p : out     std_logic;
addr1_ram1_p : out     std_logic;
addr1_ram2_p : out     std_logic;
addr1_ram3_p : out     std_logic;
addr1_ram4_p : out     std_logic;
addr1_ram5_p : out     std_logic;
addr1_ram6_p : out     std_logic;
addr1_ram7_p : out     std_logic;
addr1_ram8_p : out     std_logic;
addr1_ram9_p : out     std_logic;
addr1_ram10_p : out    std_logic;
addr1_ram11_p : out    std_logic;
addr1_ram12_p : out    std_logic;
addr1_ram13_p : out    std_logic;
addr1_ram14_p : out    std_logic;
addr1_ram15_p : out    std_logic;
addr1_ram16_p : out    std_logic;
addr1_ram17_p : out    std_logic;
```

```
data1_ram0_p      : inout    std_logic;
data1_ram1_p      : inout    std_logic;
data1_ram2_p      : inout    std_logic;
data1_ram3_p      : inout    std_logic;
data1_ram4_p      : inout    std_logic;
data1_ram5_p      : inout    std_logic;
data1_ram6_p      : inout    std_logic;
data1_ram7_p      : inout    std_logic;
data1_ram8_p      : inout    std_logic;
data1_ram9_p      : inout    std_logic;
data1_ram10_p     : inout    std_logic;
data1_ram11_p     : inout    std_logic;
data1_ram12_p     : inout    std_logic;
data1_ram13_p     : inout    std_logic;
data1_ram14_p     : inout    std_logic;
data1_ram15_p     : inout    std_logic;

ram1_we_p         : out      std_logic;
ram1_oe_p         : out      std_logic;
ram1_cs_p         : out      std_logic;

addr2_ram0_p      : out      std_logic;
addr2_ram1_p      : out      std_logic;
addr2_ram2_p      : out      std_logic;
addr2_ram3_p      : out      std_logic;
addr2_ram4_p      : out      std_logic;
addr2_ram5_p      : out      std_logic;
addr2_ram6_p      : out      std_logic;
addr2_ram7_p      : out      std_logic;
addr2_ram8_p      : out      std_logic;
addr2_ram9_p      : out      std_logic;
addr2_ram10_p     : out      std_logic;
addr2_ram11_p     : out      std_logic;
addr2_ram12_p     : out      std_logic;
addr2_ram13_p     : out      std_logic;
addr2_ram14_p     : out      std_logic;
addr2_ram15_p     : out      std_logic;
addr2_ram16_p     : out      std_logic;
addr2_ram17_p     : out      std_logic;

data2_ram0_p      : inout    std_logic;
data2_ram1_p      : inout    std_logic;
data2_ram2_p      : inout    std_logic;
data2_ram3_p      : inout    std_logic;
data2_ram4_p      : inout    std_logic;
data2_ram5_p      : inout    std_logic;
data2_ram6_p      : inout    std_logic;
data2_ram7_p      : inout    std_logic;
data2_ram8_p      : inout    std_logic;
data2_ram9_p      : inout    std_logic;
data2_ram10_p     : inout    std_logic;
data2_ram11_p     : inout    std_logic;
data2_ram12_p     : inout    std_logic;
data2_ram13_p     : inout    std_logic;
data2_ram14_p     : inout    std_logic;
data2_ram15_p     : inout    std_logic;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 21/131
Filename: *ASTEDetailedDesign6*

```
ram2_we_p      : out      std_logic;  
ram2_oe_p      : out      std_logic;  
ram2_cs_p      : out      std_logic;  
  
ser1_rx_p      : in       std_logic;  
ser1_tx_p      : out      std_logic;  
ser2_rx_p      : in       std_logic;  
ser2_tx_p      : out      std_logic;  
  
hk_dout1_p     : in       std_logic;  
hk_dout2_p     : in       std_logic;  
hk_din_p       : out      std_logic;  
sclk_p         : out      std_logic;  
cs_p           : out      std_logic );
```

```
attribute alspin : string;  
attribute alspin of ck20_p      : signal is "153";  
attribute alspin of rst_cpu_p   : signal is "207";  
attribute alspin of rst0_p      : signal is "3";  
attribute alspin of sync_p      : signal is "206";  
attribute alspin of addr_dsp0_p : signal is "158";  
attribute alspin of addr_dsp1_p : signal is "161";  
attribute alspin of addr_dsp2_p : signal is "165";  
attribute alspin of addr_dsp3_p : signal is "168";  
attribute alspin of addr_dsp4_p : signal is "171";  
attribute alspin of addr_dsp5_p : signal is "174";  
attribute alspin of addr_dsp6_p : signal is "177";  
attribute alspin of addr_dsp7_p : signal is "187";  
attribute alspin of addr_dsp8_p : signal is "190";  
attribute alspin of addr_dsp9_p : signal is "193";  
attribute alspin of addr_dsp10_p : signal is "196";  
attribute alspin of addr_dsp11_p : signal is "199";  
attribute alspin of addr_dsp12_p : signal is "203";  
attribute alspin of addr_dsp13_p : signal is "5";  
attribute alspin of addr_dsp14_p : signal is "8";  
attribute alspin of addr_dsp15_p : signal is "13";  
attribute alspin of addr_dsp16_p : signal is "16";  
attribute alspin of addr_dsp17_p : signal is "19";
```

```
attribute alspin of data_dsp8_p : signal is "66";  
attribute alspin of data_dsp9_p : signal is "70";  
attribute alspin of data_dsp10_p : signal is "74";  
attribute alspin of data_dsp11_p : signal is "84";  
attribute alspin of data_dsp12_p : signal is "88";  
attribute alspin of data_dsp13_p : signal is "92";  
attribute alspin of data_dsp14_p : signal is "96";  
attribute alspin of data_dsp15_p : signal is "101";  
attribute alspin of data_dsp16_p : signal is "24";  
attribute alspin of data_dsp17_p : signal is "33";  
attribute alspin of data_dsp18_p : signal is "37";  
attribute alspin of data_dsp19_p : signal is "43";  
attribute alspin of data_dsp20_p : signal is "47";  
attribute alspin of data_dsp21_p : signal is "51";  
attribute alspin of data_dsp22_p : signal is "56";  
attribute alspin of data_dsp23_p : signal is "61";
```

```
attribute alspin of dms1_p      : signal is "127";
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 22/131
Filename: ASTEDetailedDesign6

```
attribute alspin of dms2_p      : signal is "126";
attribute alspin of dms3_p      : signal is "125";
attribute alspin of dmrdr_p     : signal is "136";
attribute alspin of dmwr_p     : signal is "128";
attribute alspin of flag2_p     : signal is "119";
attribute alspin of flag3_p     : signal is "118";

attribute alspin of bit16_p     : signal is "124";
attribute alspin of irq3_p      : signal is "110";
attribute alspin of irq1_p      : signal is "112";
attribute alspin of irq0_p      : signal is "113";

attribute alspin of data_cam0_p : signal is "67";
attribute alspin of data_cam1_p : signal is "71";
attribute alspin of data_cam2_p : signal is "75";
attribute alspin of data_cam3_p : signal is "85";
attribute alspin of data_cam4_p : signal is "89";
attribute alspin of data_cam5_p : signal is "93";
attribute alspin of data_cam6_p : signal is "97";
attribute alspin of data_cam7_p : signal is "102";
attribute alspin of data_cam8_p : signal is "29";
attribute alspin of data_cam9_p : signal is "34";
attribute alspin of data_cam10_p : signal is "38";
attribute alspin of data_cam11_p : signal is "44";
attribute alspin of data_cam12_p : signal is "48";
attribute alspin of data_cam13_p : signal is "53";
attribute alspin of data_cam14_p : signal is "57";
attribute alspin of data_cam15_p : signal is "62";

attribute alspin of cam_we_p    : signal is "116";
attribute alspin of sel_cam_p   : signal is "123";
attribute alspin of rst_row_p   : signal is "142";
attribute alspin of end_im_p    : signal is "143";
attribute alspin of im_we_p    : signal is "141";
attribute alspin of dir_p      : signal is "135";

attribute alspin of addr1_ram0_p : signal is "160";
attribute alspin of addr1_ram1_p : signal is "163";
attribute alspin of addr1_ram2_p : signal is "167";
attribute alspin of addr1_ram3_p : signal is "170";
attribute alspin of addr1_ram4_p : signal is "173";
attribute alspin of addr1_ram5_p : signal is "176";
attribute alspin of addr1_ram6_p : signal is "179";
attribute alspin of addr1_ram7_p : signal is "189";
attribute alspin of addr1_ram8_p : signal is "192";
attribute alspin of addr1_ram9_p : signal is "195";
attribute alspin of addr1_ram10_p : signal is "198";
attribute alspin of addr1_ram11_p : signal is "202";
attribute alspin of addr1_ram12_p : signal is "205";
attribute alspin of addr1_ram13_p : signal is "7";
attribute alspin of addr1_ram14_p : signal is "10";
attribute alspin of addr1_ram15_p : signal is "15";
attribute alspin of addr1_ram16_p : signal is "18";
attribute alspin of addr1_ram17_p : signal is "21";

attribute alspin of data1_ram0_p : signal is "63";
attribute alspin of data1_ram1_p : signal is "68";
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 23/131
Filename: ASTEDetailedDesign6

```
attribute alspin of data1_ram2_p : signal is "72";
attribute alspin of data1_ram3_p : signal is "81";
attribute alspin of data1_ram4_p : signal is "86";
attribute alspin of data1_ram5_p : signal is "90";
attribute alspin of data1_ram6_p : signal is "94";
attribute alspin of data1_ram7_p : signal is "99";
attribute alspin of data1_ram8_p : signal is "22";
attribute alspin of data1_ram9_p : signal is "31";
attribute alspin of data1_ram10_p : signal is "35";
attribute alspin of data1_ram11_p : signal is "39";
attribute alspin of data1_ram12_p : signal is "45";
attribute alspin of data1_ram13_p : signal is "49";
attribute alspin of data1_ram14_p : signal is "54";
attribute alspin of data1_ram15_p : signal is "58";

attribute alspin of ram1_we_p : signal is "133";
attribute alspin of ram1_oe_p : signal is "137";
attribute alspin of ram1_cs_p : signal is "139";

attribute alspin of addr2_ram0_p : signal is "159";
attribute alspin of addr2_ram1_p : signal is "162";
attribute alspin of addr2_ram2_p : signal is "166";
attribute alspin of addr2_ram3_p : signal is "169";
attribute alspin of addr2_ram4_p : signal is "172";
attribute alspin of addr2_ram5_p : signal is "175";
attribute alspin of addr2_ram6_p : signal is "178";
attribute alspin of addr2_ram7_p : signal is "188";
attribute alspin of addr2_ram8_p : signal is "191";
attribute alspin of addr2_ram9_p : signal is "194";
attribute alspin of addr2_ram10_p : signal is "197";
attribute alspin of addr2_ram11_p : signal is "200";
attribute alspin of addr2_ram12_p : signal is "204";
attribute alspin of addr2_ram13_p : signal is "6";
attribute alspin of addr2_ram14_p : signal is "9";
attribute alspin of addr2_ram15_p : signal is "14";
attribute alspin of addr2_ram16_p : signal is "17";
attribute alspin of addr2_ram17_p : signal is "20";

attribute alspin of data2_ram0_p : signal is "64";
attribute alspin of data2_ram1_p : signal is "69";
attribute alspin of data2_ram2_p : signal is "73";
attribute alspin of data2_ram3_p : signal is "83";
attribute alspin of data2_ram4_p : signal is "87";
attribute alspin of data2_ram5_p : signal is "91";
attribute alspin of data2_ram6_p : signal is "95";
attribute alspin of data2_ram7_p : signal is "100";
attribute alspin of data2_ram8_p : signal is "23";
attribute alspin of data2_ram9_p : signal is "32";
attribute alspin of data2_ram10_p : signal is "36";
attribute alspin of data2_ram11_p : signal is "42";
attribute alspin of data2_ram12_p : signal is "46";
attribute alspin of data2_ram13_p : signal is "50";
attribute alspin of data2_ram14_p : signal is "55";
attribute alspin of data2_ram15_p : signal is "59";

attribute alspin of ram2_we_p : signal is "134";
attribute alspin of ram2_oe_p : signal is "138";
```

```

attribute alspin of ram2_cs_p      : signal is "140";

attribute alspin of ser1_rx_p     : signal is "144";
attribute alspin of ser1_tx_p     : signal is "147";
attribute alspin of ser2_rx_p     : signal is "149";
attribute alspin of ser2_tx_p     : signal is "150";

attribute alspin of hk_dout1_p    : signal is "152";
attribute alspin of hk_dout2_p    : signal is "156";
attribute alspin of hk_din_p      : signal is "151";
attribute alspin of sclk_p        : signal is "154";
attribute alspin of cs_p          : signal is "155";

attribute alspin of strb_p        : signal is "106";
attribute alspin of stx_p         : signal is "107";

```

end top_level;

architecture arc_top_level of top_level is

```

component dsp_data
  port ( addr_dsp      : in      std_logic_vector (5 downto 4);
        ser1_r        : in      std_logic_vector (7 downto 0);
        ser2_r        : in      std_logic_vector (7 downto 0);
        timer         : in      std_logic_vector (15 downto 0);
        hk_data       : in      std_logic_vector (15 downto 0);
        ram_im_r      : in      std_logic_vector (15 downto 0);
        data_dsp_out  : out     std_logic_vector (15 downto 0);
        dms2          : in      std_logic );
end component;

component mux_addr
  port ( flag2        : in      std_logic;
        ck            : in      std_logic;
        rst           : in      std_logic;
        addr_dsp      : in      std_logic_vector (17 downto 0);
        addr_cam      : in      std_logic_vector (17 downto 0);
        addr1_ram     : out     std_logic_vector (17 downto 0);
        addr2_ram     : out     std_logic_vector (17 downto 0) );
end component;

component mux_data
  port ( flag2        : in      std_logic;
        ck            : in      std_logic;
        rst           : in      std_logic;
        data_dsp_in   : in      std_logic_vector (15 downto 0);
        ram_im_r      : out     std_logic_vector (15 downto 0);
        data_cam_in   : in      std_logic_vector (15 downto 0);
        data1_ram_out : out     std_logic_vector (15 downto 0);
        data1_ram_in  : in      std_logic_vector (15 downto 0);
        data2_ram_out : out     std_logic_vector (15 downto 0);
        data2_ram_in  : in      std_logic_vector (15 downto 0) );
end component;

component counter_addr

```

```

port ( rst           : in      std_logic;
      end_im        : in      std_logic;
      rst_row       : in      std_logic;
      im_we         : in      std_logic;
      addr_cam      : out     std_logic_vector (17 downto 0) );
end component;

component serial
port ( rst           : in      std_logic;
      ck20          : in      std_logic;
      ser_rx        : in      std_logic;
      ser_tx        : out     std_logic;
      data_dsp_in   : in      std_logic_vector (7 downto 0);
      ser_r         : out     std_logic_vector (7 downto 0);
      addr_dsp      : in      std_logic_vector (2 downto 0);
      inttx         : out     std_logic;
      intrx         : out     std_logic;
      write_ser     : in      std_logic;
      read_ser      : in      std_logic;
      dms3          : in      std_logic );
end component;

component timer_data
port ( rst           : in      std_logic;
      ck20          : in      std_logic;
      sync          : in      std_logic;
      addr_dsp      : in      std_logic_vector (0 downto 0);
      timer         : out     std_logic_vector (15 downto 0) );
end component;

component hk
port ( stx          : out     std_logic;
      strb         : out     std_logic;
      rst          : in      std_logic;
      hk_dout1     : in      std_logic;
      hk_dout2     : in      std_logic;
      hk_din       : out     std_logic;
      cs           : out     std_logic;
      ck           : in      std_logic;
      sclk        : out     std_logic;
      hk_data      : out     std_logic_vector (15 downto 0);
      addr_dsp     : in      std_logic_vector (3 downto 0);
      dms3         : in      std_logic;
      irq0         : out     std_logic );
end component;

component bus_cont
port ( rst           : in      std_logic;
      ck            : in      std_logic;
      dms1         : in      std_logic;
      dms2         : in      std_logic;
      dms3         : in      std_logic;
      dmrd         : in      std_logic;
      dmwr         : in      std_logic;
      flag2        : in      std_logic;
      im_we        : in      std_logic;
      data_dsp_in  : in      std_logic_vector (7 downto 0);

```

```

addr_dsp      : in      std_logic_vector (5 downto 4);
end_im        : in      std_logic;
inttx1        : in      std_logic;
intrx1        : in      std_logic;
inttx2        : in      std_logic;
intrx2        : in      std_logic;
sel_data_dsp  : out      std_logic;
sel_data_cam  : out      std_logic;
sel_data_camx : out      std_logic;
sel_cam       : out      std_logic;
bit16         : out      std_logic;
irq1          : out      std_logic;
irq3          : out      std_logic;
dir           : out      std_logic;
ram1_we       : out      std_logic;
ram1_oe       : out      std_logic;
ram1_cs       : out      std_logic;
ram2_we       : out      std_logic;
ram2_oe       : out      std_logic;
ram2_cs       : out      std_logic;
write_ser1    : out      std_logic;
read_ser1     : out      std_logic;
write_ser2    : out      std_logic;
read_ser2     : out      std_logic;
cam_we        : out      std_logic );
end component;

component timeout
port ( rst      : in      std_logic;
      ck        : in      std_logic;
      flag3     : in      std_logic;
      rst_cpu   : out     std_logic );
end component;

component inbuf
port ( pad      : in      std_logic := 'U';
      y         : out     std_logic );
end component;

component outbuf
port ( d        : in      std_logic := 'U';
      pad       : out     std_logic );
end component;

component bibuf
port ( pad      : inout   std_logic;
      d         : in      std_logic := 'U';
      e         : in      std_logic := 'U';
      y         : out     std_logic );
end component;

component tribuff
port ( d        : in      std_logic := 'U';
      e         : in      std_logic := 'U';
      pad       : out     std_logic );
end component;

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 27/131
Filename: ASTEDetailedDesign6

```
signal ck20          : std_logic;
signal rst           : std_logic;
signal rst0          : std_logic;
signal rst_cpu       : std_logic;
signal sync          : std_logic;
signal addr_dsp      : std_logic_vector (17 downto 0);
signal data_dsp      : std_logic_vector (15 downto 0);
signal data_dsp_in   : std_logic_vector (15 downto 0);
signal data_dsp_out  : std_logic_vector (15 downto 0);
signal ram_im_r      : std_logic_vector (15 downto 0);
signal ser1_r        : std_logic_vector (7 downto 0);
signal ser2_r        : std_logic_vector (7 downto 0);
signal timer         : std_logic_vector (15 downto 0);
signal hk_data       : std_logic_vector (15 downto 0);
signal addr_cam      : std_logic_vector (17 downto 0);
signal dms1          : std_logic;
signal dms2          : std_logic;
signal dms3          : std_logic;
signal dmr           : std_logic;
signal dmwr         : std_logic;
signal flag2         : std_logic;
signal flag3         : std_logic;
signal data_cam      : std_logic_vector (15 downto 0);
signal data_cam_in   : std_logic_vector (15 downto 0);
signal data_cam_out  : std_logic_vector (15 downto 0);
signal cam_we        : std_logic;
signal sel_cam       : std_logic;
signal rst_row       : std_logic;
signal end_im        : std_logic;
signal im_we         : std_logic;
signal dir           : std_logic;
signal addr1_ram     : std_logic_vector (17 downto 0);
signal data1_ram     : std_logic_vector (15 downto 0);
signal data1_ram_in  : std_logic_vector (15 downto 0);
signal data1_ram_out : std_logic_vector (15 downto 0);
signal ram1_we       : std_logic;
signal ram1_oe       : std_logic;
signal ram1_cs       : std_logic;
signal addr2_ram     : std_logic_vector (17 downto 0);
signal data2_ram     : std_logic_vector (15 downto 0);
signal data2_ram_in  : std_logic_vector (15 downto 0);
signal data2_ram_out : std_logic_vector (15 downto 0);
signal ram2_we       : std_logic;
signal ram2_oe       : std_logic;
signal ram2_cs       : std_logic;

signal bit16         : std_logic;
signal irq0          : std_logic;
signal irq1          : std_logic;
signal irq3          : std_logic;
signal ser1_rx       : std_logic;
signal ser1_tx       : std_logic;
signal ser2_rx       : std_logic;
signal ser2_tx       : std_logic;
signal write_ser1    : std_logic;
signal write_ser2    : std_logic;
signal read_ser1     : std_logic;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 28/131
Filename: *ASTEDetailedDesign6*

```
signal read_ser2      : std_logic;

signal inttx1         : std_logic;
signal intrx1         : std_logic;
signal inttx2         : std_logic;
signal intrx2         : std_logic;

signal sel_data_cam   : std_logic;
signal sel_data_camx  : std_logic;
signal sel_data_dsp   : std_logic;

signal hk_dout1       : std_logic;
signal hk_dout2       : std_logic;
signal hk_din         : std_logic;
signal cs             : std_logic;
signal sclk           : std_logic;
signal dm             : std_logic;

signal strb           : std_logic;
signal stx            : std_logic;
```

begin

```
inst1 : dsp_data
  port map ( addr_dsp(5 downto 4)=>addr_dsp(5 downto 4),
            ser1_r(7 downto 0)=>ser1_r(7 downto 0),
            ser2_r(7 downto 0)=>ser2_r(7 downto 0),
            timer(15 downto 0)=>timer(15 downto 0),
            hk_data(15 downto 0)=>hk_data(15 downto 0),
            ram_im_r(15 downto 0)=>ram_im_r(15 downto 0),
            data_dsp_out(15 downto 0)=>data_dsp_out(15 downto 0),
            dms2=>dms2 );

inst2 : mux_addr
  port map ( flag2=>flag2, ck=>ck20, rst=>rst,
            addr_dsp(17 downto 0)=>addr_dsp(17 downto 0),
            addr_cam(17 downto 0)=>addr_cam(17 downto 0),
            addr1_ram(17 downto 0)=>addr1_ram(17 downto 0),
            addr2_ram(17 downto 0)=>addr2_ram(17 downto 0) );

inst3 : mux_data
  port map ( flag2=>flag2, ck=>ck20, rst=>rst,
            data_dsp_in(15 downto 0)=>data_dsp(15 downto 0),
            ram_im_r(15 downto 0)=>ram_im_r(15 downto 0),
            data_cam_in(15 downto 0)=>data_cam(15 downto 0),
            data1_ram_out(15 downto 0)=>data1_ram_out(15 downto 0),
            data1_ram_in(15 downto 0)=>data1_ram(15 downto 0),
            data2_ram_out(15 downto 0)=>data2_ram_out(15 downto 0),
            data2_ram_in(15 downto 0)=>data2_ram(15 downto 0) );

inst4 : counter_addr
  port map ( rst=>rst, end_im=>end_im, rst_row=>rst_row, im_we=>im_we,
            addr_cam(17 downto 0)=>addr_cam(17 downto 0) );

inst4a : serial
  port map ( rst=>rst, ck20=>ck20, ser_rx=>ser1_rx, ser_tx=>ser1_tx,
            data_dsp_in(7 downto 0)=>data_dsp_in(7 downto 0),
            ser_r(7 downto 0)=>ser1_r(7 downto 0),
            addr_dsp(2 downto 0)=>addr_dsp(2 downto 0),
            inttx=>inttx1, intrx=>intrx1,
```

```

        write_ser=>write_ser1, read_ser=>read_ser1,
        dms3=>dms3 );
inst4b : serial
    port map ( rst=>rst, ck20=>ck20, ser_rx=>ser2_rx, ser_tx=>ser2_tx,
        data_dsp_in(7 downto 0)=>data_dsp_in(7 downto 0),
        ser_r(7 downto 0)=>ser2_r(7 downto 0),
        addr_dsp(2 downto 0)=>addr_dsp(2 downto 0),
        inttx=>inttx2, intrx=>intrx2,
        write_ser=>write_ser2, read_ser=>read_ser2,
        dms3=>dms3 );
inst6s : timer_data
    port map ( rst=>rst, ck20=>ck20, sync=>sync,
        addr_dsp(0 downto 0)=>addr_dsp(0 downto 0),
        timer(15 downto 0)=>timer(15 downto 0) );
inst7s : hk
    port map ( stx=>stx, strb=>strb,
        rst=>rst,
        hk_dout1=>hk_dout1,
        hk_dout2=>hk_dout2,
        hk_din=>hk_din, cs=>cs, ck=>ck20, sclk=>sclk,
        hk_data(15 downto 0)=>hk_data(15 downto 0),
        addr_dsp(3 downto 0)=>addr_dsp(3 downto 0),
        dms3=>dms3, irq0=>irq0 );
inst5 : bus_cont
    port map ( rst=>rst, ck=>ck20, dms1=>dms1, dms2=>dms2, dms3=>dms3,
dmr=>dmr,
        dmwr=>dmwr, flag2=>flag2, im_we=>im_we,
        data_dsp_in(7 downto 0)=>data_dsp_in(7 downto 0),
        addr_dsp(5 downto 4)=>addr_dsp(5 downto 4),
        end_im=>end_im, inttx1=>inttx1, intrx1=>intrx1,
        inttx2=>inttx2, intrx2=>intrx2,
        sel_data_dsp=>sel_data_dsp,
        sel_data_cam=>sel_data_cam, sel_data_camx=>sel_data_camx,
        sel_cam=>sel_cam,
        bit16=>bit16, irq1=>irq1, irq3=>irq3, dir=>dir,
        ram1_we=>ram1_we, ram1_oe=>ram1_oe, ram1_cs=>ram1_cs,
        ram2_we=>ram2_we, ram2_oe=>ram2_oe, ram2_cs=>ram2_cs,
        write_ser1=>write_ser1, read_ser1=>read_ser1,
        write_ser2=>write_ser2, read_ser2=>read_ser2,
        cam_we=>cam_we );
inst8s : timeout
    port map ( rst=>rst, ck=>ck20, flag3 =>flag3, rst_cpu=>rst_cpu );
inst6 : inbuf
    port map ( pad=>ck20_p, y=>ck20 );
inst7a : inbuf
    port map ( pad=>rst0_p , y=>rst0 );
inst7 : tribuff
    port map ( d=>'0', e=>rst_cpu, pad=>rst_cpu_p );
inst7b : inbuf
    port map ( pad=>sync_p , y=>sync );
inst8 : inbuf
    port map ( pad=>addr_dsp0_p , y=>addr_dsp(0) );

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 30/131
Filename: *ASTEDetailedDesign6*

```
inst9 : inbuf
  port map ( pad=>addr_dsp1_p , y=>addr_dsp(1) );
inst10 : inbuf
  port map ( pad=>addr_dsp2_p , y=>addr_dsp(2) );
inst11 : inbuf
  port map ( pad=>addr_dsp3_p , y=>addr_dsp(3) );
inst12 : inbuf
  port map ( pad=>addr_dsp4_p , y=>addr_dsp(4) );
inst13 : inbuf
  port map ( pad=>addr_dsp5_p , y=>addr_dsp(5) );
inst14 : inbuf
  port map ( pad=>addr_dsp6_p , y=>addr_dsp(6) );
inst15 : inbuf
  port map ( pad=>addr_dsp7_p , y=>addr_dsp(7) );
inst16 : inbuf
  port map ( pad=>addr_dsp8_p , y=>addr_dsp(8) );
inst17 : inbuf
  port map ( pad=>addr_dsp9_p , y=>addr_dsp(9) );
inst18 : inbuf
  port map ( pad=>addr_dsp10_p , y=>addr_dsp(10) );
inst19 : inbuf
  port map ( pad=>addr_dsp11_p , y=>addr_dsp(11) );
inst20 : inbuf
  port map ( pad=>addr_dsp12_p , y=>addr_dsp(12) );
inst21 : inbuf
  port map ( pad=>addr_dsp13_p , y=>addr_dsp(13) );
inst22 : inbuf
  port map ( pad=>addr_dsp14_p , y=>addr_dsp(14) );
inst23 : inbuf
  port map ( pad=>addr_dsp15_p , y=>addr_dsp(15) );
inst24 : inbuf
  port map ( pad=>addr_dsp16_p , y=>addr_dsp(16) );
inst25 : inbuf
  port map ( pad=>addr_dsp17_p , y=>addr_dsp(17) );

inst26 : bibuf
  port map ( pad=>data_dsp8_p, d=>data_dsp_out(0),
            e=>sel_data_dsp, y=>data_dsp_in(0) );
inst27 : bibuf
  port map ( pad=>data_dsp9_p, d=>data_dsp_out(1),
            e=>sel_data_dsp, y=>data_dsp_in(1) );
inst28 : bibuf
  port map ( pad=>data_dsp10_p, d=>data_dsp_out(2),
            e=>sel_data_dsp, y=>data_dsp_in(2) );
inst29 : bibuf
  port map ( pad=>data_dsp11_p, d=>data_dsp_out(3),
            e=>sel_data_dsp, y=>data_dsp_in(3) );
inst30 : bibuf
  port map ( pad=>data_dsp12_p, d=>data_dsp_out(4),
            e=>sel_data_dsp, y=>data_dsp_in(4) );
inst31 : bibuf
  port map ( pad=>data_dsp13_p, d=>data_dsp_out(5),
            e=>sel_data_dsp, y=>data_dsp_in(5) );
inst32 : bibuf
  port map ( pad=>data_dsp14_p, d=>data_dsp_out(6),
            e=>sel_data dsp, y=>data dsp in(6) );
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 31/131
Filename: *ASTEDetailedDesign6*

```
inst33 : bibuf
  port map ( pad=>data_dsp15_p, d=>data_dsp_out(7),
            e=>sel_data_dsp, y=>data_dsp_in(7) );
inst34 : bibuf
  port map ( pad=>data_dsp16_p, d=>data_dsp_out(8),
            e=>sel_data_dsp, y=>data_dsp_in(8) );
inst35 : bibuf
  port map ( pad=>data_dsp17_p, d=>data_dsp_out(9),
            e=>sel_data_dsp, y=>data_dsp_in(9) );
inst36 : bibuf
  port map ( pad=>data_dsp18_p, d=>data_dsp_out(10),
            e=>sel_data_dsp, y=>data_dsp_in(10) );
inst37 : bibuf
  port map ( pad=>data_dsp19_p, d=>data_dsp_out(11),
            e=>sel_data_dsp, y=>data_dsp_in(11) );
inst38 : bibuf
  port map ( pad=>data_dsp20_p, d=>data_dsp_out(12),
            e=>sel_data_dsp, y=>data_dsp_in(12) );
inst39 : bibuf
  port map ( pad=>data_dsp21_p, d=>data_dsp_out(13),
            e=>sel_data_dsp, y=>data_dsp_in(13) );
inst40 : bibuf
  port map ( pad=>data_dsp22_p, d=>data_dsp_out(14),
            e=>sel_data_dsp, y=>data_dsp_in(14) );
inst41 : bibuf
  port map ( pad=>data_dsp23_p, d=>data_dsp_out(15),
            e=>sel_data_dsp, y=>data_dsp_in(15) );

inst41s : inbuf
  port map ( pad=>dms1_p , y=>dms1 );
inst42 : inbuf
  port map ( pad=>dms2_p , y=>dms2 );
inst43 : inbuf
  port map ( pad=>dms3_p , y=>dms3 );
inst44 : inbuf
  port map ( pad=>dms4_p , y=>dms4 );
inst45 : inbuf
  port map ( pad=>dmwr_p , y=>dmwr );
inst46 : inbuf
  port map ( pad=>flag2_p , y=>flag2 );
inst46s : inbuf
  port map ( pad=>flag3_p , y=>flag3 );

inst47 : bibuf
  port map ( pad=>data_cam0_p, d=>data_cam_out(0),
            e=>sel_data_cam, y=>data_cam_in(0) );
inst48 : bibuf
  port map ( pad=>data_cam1_p, d=>data_cam_out(1),
            e=>sel_data_cam, y=>data_cam_in(1) );
inst49 : bibuf
  port map ( pad=>data_cam2_p, d=>data_cam_out(2),
            e=>sel_data_cam, y=>data_cam_in(2) );
inst50 : bibuf
  port map ( pad=>data_cam3_p, d=>data_cam_out(3),
            e=>sel_data_cam, y=>data_cam_in(3) );
inst51 : bibuf
  port map ( pad=>data_cam4_p, d=>data_cam_out(4),
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 32/131
Filename: ASTEDetailedDesign6

```

        e=>sel_data_cam, y=>data_cam_in(4) );
inst52 : bibuf
    port map ( pad=>data_cam5_p, d=>data_cam_out(5),
        e=>sel_data_cam, y=>data_cam_in(5) );
inst53 : bibuf
    port map ( pad=>data_cam6_p, d=>data_cam_out(6),
        e=>sel_data_cam, y=>data_cam_in(6) );
inst54 : bibuf
    port map ( pad=>data_cam7_p, d=>data_cam_out(7),
        e=>sel_data_cam, y=>data_cam_in(7) );
inst55 : bibuf
    port map ( pad=>data_cam8_p, d=>data_cam_out(8),
        e=>sel_data_cam, y=>data_cam_in(8) );
inst56 : bibuf
    port map ( pad=>data_cam9_p, d=>data_cam_out(9),
        e=>sel_data_cam, y=>data_cam_in(9) );
inst57 : bibuf
    port map ( pad=>data_cam10_p, d=>data_cam_out(10),
        e=>sel_data_cam, y=>data_cam_in(10) );
inst58 : bibuf
    port map ( pad=>data_cam11_p, d=>data_cam_out(11),
        e=>sel_data_cam, y=>data_cam_in(11) );
inst59 : bibuf
    port map ( pad=>data_cam12_p, d=>data_cam_out(12),
        e=>sel_data_cam, y=>data_cam_in(12) );
inst60 : bibuf
    port map ( pad=>data_cam13_p, d=>data_cam_out(13),
        e=>sel_data_cam, y=>data_cam_in(13) );
inst61 : bibuf
    port map ( pad=>data_cam14_p, d=>data_cam_out(14),
        e=>sel_data_cam, y=>data_cam_in(14) );
inst62 : bibuf
    port map ( pad=>data_cam15_p, d=>data_cam_out(15),
        e=>sel_data_cam, y=>data_cam_in(15) );

inst62s : outbuf
    port map ( d=>cam_we, pad=>cam_we_p );
inst63 : outbuf
    port map ( d=>sel_cam, pad=>sel_cam_p );

inst63s : inbuf
    port map ( pad=>rst_row_p, y=>rst_row );
inst64 : inbuf
    port map ( pad=>end_im_p, y=>end_im );
inst65 : inbuf
    port map ( pad=>im_we_p, y=>im_we );

inst66 : outbuf
    port map ( d=>dir, pad=>dir_p );

inst67 : outbuf
    port map ( d=>addr1_ram(0), pad=>addr1_ram0_p );
inst68 : outbuf
    port map ( d=>addr1_ram(1), pad=>addr1_ram1_p );
inst69 : outbuf
    port map ( d=>addr1_ram(2), pad=>addr1_ram2_p );
inst70 : outbuf
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 33/131
Filename: *ASTEDetailedDesign6*

```
port map ( d=>addr1_ram(3), pad=>addr1_ram3_p );
inst71 : outbuf
port map ( d=>addr1_ram(4), pad=>addr1_ram4_p );
inst72 : outbuf
port map ( d=>addr1_ram(5), pad=>addr1_ram5_p );
inst73 : outbuf
port map ( d=>addr1_ram(6), pad=>addr1_ram6_p );
inst74 : outbuf
port map ( d=>addr1_ram(7), pad=>addr1_ram7_p );
inst75 : outbuf
port map ( d=>addr1_ram(8), pad=>addr1_ram8_p );
inst76 : outbuf
port map ( d=>addr1_ram(9), pad=>addr1_ram9_p );
inst77 : outbuf
port map ( d=>addr1_ram(10), pad=>addr1_ram10_p );
inst78 : outbuf
port map ( d=>addr1_ram(11), pad=>addr1_ram11_p );
inst79 : outbuf
port map ( d=>addr1_ram(12), pad=>addr1_ram12_p );
inst80 : outbuf
port map ( d=>addr1_ram(13), pad=>addr1_ram13_p );
inst81 : outbuf
port map ( d=>addr1_ram(14), pad=>addr1_ram14_p );
inst82 : outbuf
port map ( d=>addr1_ram(15), pad=>addr1_ram15_p );
inst83 : outbuf
port map ( d=>addr1_ram(16), pad=>addr1_ram16_p );
inst84 : outbuf
port map ( d=>addr1_ram(17), pad=>addr1_ram17_p );

inst85 : bibuf
port map ( pad=>data1_ram0_p, d=>data1_ram_out(0),
          e=>ram1_oe, y=>data1_ram_in(0) );
inst86 : bibuf
port map ( pad=>data1_ram1_p, d=>data1_ram_out(1),
          e=>ram1_oe, y=>data1_ram_in(1) );
inst87 : bibuf
port map ( pad=>data1_ram2_p, d=>data1_ram_out(2),
          e=>ram1_oe, y=>data1_ram_in(2) );
inst88 : bibuf
port map ( pad=>data1_ram3_p, d=>data1_ram_out(3),
          e=>ram1_oe, y=>data1_ram_in(3) );
inst89 : bibuf
port map ( pad=>data1_ram4_p, d=>data1_ram_out(4),
          e=>ram1_oe, y=>data1_ram_in(4) );
inst90 : bibuf
port map ( pad=>data1_ram5_p, d=>data1_ram_out(5),
          e=>ram1_oe, y=>data1_ram_in(5) );
inst91 : bibuf
port map ( pad=>data1_ram6_p, d=>data1_ram_out(6),
          e=>ram1_oe, y=>data1_ram_in(6) );
inst92 : bibuf
port map ( pad=>data1_ram7_p, d=>data1_ram_out(7),
          e=>ram1_oe, y=>data1_ram_in(7) );
inst93 : bibuf
port map ( pad=>data1_ram8_p, d=>data1_ram_out(8),
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 34/131
Filename: *ASTEDetailedDesign6*

```

        e=>ram1_oe, y=>data1_ram_in(8) );
inst94 : bibuf
    port map ( pad=>data1_ram9_p, d=>data1_ram_out(9),
              e=>ram1_oe, y=>data1_ram_in(9) );
inst95 : bibuf
    port map ( pad=>data1_ram10_p, d=>data1_ram_out(10),
              e=>ram1_oe, y=>data1_ram_in(10) );
inst96 : bibuf
    port map ( pad=>data1_ram11_p, d=>data1_ram_out(11),
              e=>ram1_oe, y=>data1_ram_in(11) );
inst97 : bibuf
    port map ( pad=>data1_ram12_p, d=>data1_ram_out(12),
              e=>ram1_oe, y=>data1_ram_in(12) );
inst98 : bibuf
    port map ( pad=>data1_ram13_p, d=>data1_ram_out(13),
              e=>ram1_oe, y=>data1_ram_in(13) );
inst99 : bibuf
    port map ( pad=>data1_ram14_p, d=>data1_ram_out(14),
              e=>ram1_oe, y=>data1_ram_in(14) );
inst100 : bibuf
    port map ( pad=>data1_ram15_p, d=>data1_ram_out(15),
              e=>ram1_oe, y=>data1_ram_in(15) );

inst101 : outbuf
    port map ( d=>ram1_we, pad=>ram1_we_p );
inst102 : outbuf
    port map ( d=>ram1_oe, pad=>ram1_oe_p );
inst103 : outbuf
    port map ( d=>ram1_cs, pad=>ram1_cs_p );

inst104 : outbuf
    port map ( d=>addr2_ram(0), pad=>addr2_ram0_p );
inst105 : outbuf
    port map ( d=>addr2_ram(1), pad=>addr2_ram1_p );
inst106 : outbuf
    port map ( d=>addr2_ram(2), pad=>addr2_ram2_p );
inst107 : outbuf
    port map ( d=>addr2_ram(3), pad=>addr2_ram3_p );
inst108 : outbuf
    port map ( d=>addr2_ram(4), pad=>addr2_ram4_p );
inst109 : outbuf
    port map ( d=>addr2_ram(5), pad=>addr2_ram5_p );
inst110 : outbuf
    port map ( d=>addr2_ram(6), pad=>addr2_ram6_p );
inst111 : outbuf
    port map ( d=>addr2_ram(7), pad=>addr2_ram7_p );
inst112 : outbuf
    port map ( d=>addr2_ram(8), pad=>addr2_ram8_p );
inst113 : outbuf
    port map ( d=>addr2_ram(9), pad=>addr2_ram9_p );
inst114 : outbuf
    port map ( d=>addr2_ram(10), pad=>addr2_ram10_p );
inst115 : outbuf
    port map ( d=>addr2_ram(11), pad=>addr2_ram11_p );
inst116 : outbuf
    port map ( d=>addr2_ram(12), pad=>addr2_ram12_p );
inst117 : outbuf
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 35/131
Filename: ASTEDetailedDesign6

```
port map ( d=>addr2_ram(13), pad=>addr2_ram13_p );
inst118 : outbuf
port map ( d=>addr2_ram(14), pad=>addr2_ram14_p );
inst119 : outbuf
port map ( d=>addr2_ram(15), pad=>addr2_ram15_p );
inst120 : outbuf
port map ( d=>addr2_ram(16), pad=>addr2_ram16_p );
inst121 : outbuf
port map ( d=>addr2_ram(17), pad=>addr2_ram17_p );

inst122 : bibuf
port map ( pad=>data2_ram0_p, d=>data2_ram_out(0),
          e=>ram2_oe, y=>data2_ram_in(0) );
inst123 : bibuf
port map ( pad=>data2_ram1_p, d=>data2_ram_out(1),
          e=>ram2_oe, y=>data2_ram_in(1) );
inst124 : bibuf
port map ( pad=>data2_ram2_p, d=>data2_ram_out(2),
          e=>ram2_oe, y=>data2_ram_in(2) );
inst125 : bibuf
port map ( pad=>data2_ram3_p, d=>data2_ram_out(3),
          e=>ram2_oe, y=>data2_ram_in(3) );
inst126 : bibuf
port map ( pad=>data2_ram4_p, d=>data2_ram_out(4),
          e=>ram2_oe, y=>data2_ram_in(4) );
inst127 : bibuf
port map ( pad=>data2_ram5_p, d=>data2_ram_out(5),
          e=>ram2_oe, y=>data2_ram_in(5) );
inst128 : bibuf
port map ( pad=>data2_ram6_p, d=>data2_ram_out(6),
          e=>ram2_oe, y=>data2_ram_in(6) );
inst129 : bibuf
port map ( pad=>data2_ram7_p, d=>data2_ram_out(7),
          e=>ram2_oe, y=>data2_ram_in(7) );
inst130 : bibuf
port map ( pad=>data2_ram8_p, d=>data2_ram_out(8),
          e=>ram2_oe, y=>data2_ram_in(8) );
inst131 : bibuf
port map ( pad=>data2_ram9_p, d=>data2_ram_out(9),
          e=>ram2_oe, y=>data2_ram_in(9) );
inst132 : bibuf
port map ( pad=>data2_ram10_p, d=>data2_ram_out(10),
          e=>ram2_oe, y=>data2_ram_in(10) );
inst133 : bibuf
port map ( pad=>data2_ram11_p, d=>data2_ram_out(11),
          e=>ram2_oe, y=>data2_ram_in(11) );
inst134 : bibuf
port map ( pad=>data2_ram12_p, d=>data2_ram_out(12),
          e=>ram2_oe, y=>data2_ram_in(12) );
inst135 : bibuf
port map ( pad=>data2_ram13_p, d=>data2_ram_out(13),
          e=>ram2_oe, y=>data2_ram_in(13) );
inst136 : bibuf
port map ( pad=>data2_ram14_p, d=>data2_ram_out(14),
          e=>ram2_oe, y=>data2_ram_in(14) );
inst137 : bibuf
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 36/131
Filename: *ASTEDetailedDesign6*

```
port map ( pad=>data2_ram15_p, d=>data2_ram_out(15),
          e=>ram2_oe, y=>data2_ram_in(15) );

inst138 : outbuf
  port map ( d=>ram2_we, pad=>ram2_we_p );
inst139 : outbuf
  port map ( d=>ram2_oe, pad=>ram2_oe_p );
inst140 : outbuf
  port map ( d=>ram2_cs, pad=>ram2_cs_p );

inst141 : outbuf
  port map ( d=>bit16, pad=>bit16_p );
inst142 : outbuf
  port map ( d=>irq3, pad=>irq3_p );
inst144 : outbuf
  port map ( d=>irq1, pad=>irq1_p );
inst144s : outbuf
  port map ( d=>irq0, pad=>irq0_p );

inst145 : inbuf
  port map ( pad=>ser1_rx_p, y=>ser1_rx );
inst146 : outbuf
  port map ( d=>ser1_tx, pad=>ser1_tx_p );
inst147 : inbuf
  port map ( pad=>ser2_rx_p, y=>ser2_rx );
inst148 : outbuf
  port map ( d=>ser2_tx, pad=>ser2_tx_p );

inst150 : inbuf
  port map ( pad=>hk_dout1_p , y=>hk_dout1 );
inst152 : inbuf
  port map ( pad=>hk_dout2_p , y=>hk_dout2 );

inst154 : outbuf
  port map ( d=>sclk, pad=>sclk_p );
inst155 : outbuf
  port map ( d=>hk_din, pad=>hk_din_p );
inst156 : outbuf
  port map ( d=>cs, pad=>cs_p );

inst160 : outbuf
  port map ( d=>strb, pad=>strb_p );

inst161 : outbuf
  port map ( d=>stx, pad=>stx_p );

rst <= not rst0;

dm <= not dms2 or not dms3;
data_dsp <= data_dsp_in when dm = '1' else (others => '0');
data_cam <= data_cam_in when sel_data_camx = '0' else (others => '0');
data_cam_out <= data_dsp_in when sel_data_camx = '1' else (others => '0');
data1_ram <= data1_ram_in when dms3 = '1' else (others => '0');
data2_ram <= data2_ram_in when dms3 = '1' else (others => '0');
```

```
end arc_top_level;
```

5.1.2. dsp_data.vhd

```
-- dsp_data.vhd

library ieee;
use ieee.std_logic_1164.all;

entity dsp_data is
    port ( addr_dsp      : in      std_logic_vector (5 downto 4);
           ser1_r       : in      std_logic_vector (7 downto 0);
           ser2_r       : in      std_logic_vector (7 downto 0);
           timer        : in      std_logic_vector (15 downto 0);
           hk_data      : in      std_logic_vector (15 downto 0);
           ram_im_r     : in      std_logic_vector (15 downto 0);
           data_dsp_out : out     std_logic_vector (15 downto 0);
           dms2         : in      std_logic );
end dsp_data;

architecture arc_dsp_data of dsp_data is

    signal sel          : std_logic_vector (1 downto 0);
    signal q1           : std_logic_vector (15 downto 0);
    signal q2           : std_logic_vector (15 downto 0);

begin

    process (sel, ser1_r, ser2_r, timer, hk_data)
    begin
        case sel is
            when "00" => q1 <= "00000000" & ser1_r;
            when "01" => q1 <= "00000000" & ser2_r;
            when "10" => q1 <= timer;
            when "11" => q1 <= hk_data;
            when others => q1 <= q1;
        end case;
    end process;

    process (dms2, q1, ram_im_r)
    begin
        case dms2 is
            when '0' => q2 <= ram_im_r;
            when '1' => q2 <= q1;
            when others => q2 <= q2;
        end case;
    end process;

    -- sel(0) <= addr_dsp(4) and not dms3 ;
    -- sel(1) <= addr_dsp(5) and not dms3 ;
    sel(0) <= addr_dsp(4);
    sel(1) <= addr_dsp(5);
```

```
data_dsp_out <= q2;
```

```
end arc_dsp_data;
```

5.1.3. mux_add.vhd

```
-- mux_addr.vhd
```

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
entity mux_addr is
```

```
    port ( flag2           : in      std_logic;
           ck              : in      std_logic;
           rst             : in      std_logic;
           addr_dsp        : in      std_logic_vector (17 downto 0);
           addr_cam        : in      std_logic_vector (17 downto 0);
           addr1_ram       : out     std_logic_vector (17 downto 0);
           addr2_ram       : out     std_logic_vector (17 downto 0) );
```

```
end mux_addr;
```

```
architecture arc_mux_addr of mux_addr is
```

```
    signal q1      : std_logic_vector (17 downto 0);
    signal q2      : std_logic_vector (17 downto 0);
    signal flag2in : std_logic;
```

```
begin
```

```
    process (flag2in, addr_dsp, addr_cam)
```

```
    begin
```

```
        case flag2in is
```

```
            when '0' => q1 <= addr_dsp;
```

```
            when '1' => q1 <= addr_cam;
```

```
            when others => q1 <= q1;
```

```
        end case;
```

```
    end process;
```

```
    process (flag2in, addr_dsp, addr_cam)
```

```
    begin
```

```
        case flag2in is
```

```
            when '0' => q2 <= addr_cam;
```

```
            when '1' => q2 <= addr_dsp;
```

```
            when others => q2 <= q2;
```

```
        end case;
```

```
    end process;
```

```
    addr1_ram <= q1;
```

```
    addr2_ram <= q2;
```

```
    process(rst, ck)
```

```
    begin
```

```
        if rst = '1' then
```

```
            flag2in <= '0';
```

```
    elsif ck'event and ck = '1' then
        flag2in <= flag2;
    end if;
end process;
```

```
end arc_mux_addr;
```

5.1.4. mux_data.vhd

```
-- mux_data.vhd
```

```
library ieee;
use ieee.std_logic_1164.all;
```

```
entity mux_data is
    port ( flag2          : in      std_logic;
           ck            : in      std_logic;
           rst           : in      std_logic;
           data_dsp_in   : in      std_logic_vector (15 downto 0);
           ram_im_r      : out     std_logic_vector (15 downto 0);
           data_cam_in   : in      std_logic_vector (15 downto 0);
           data1_ram_out : out     std_logic_vector (15 downto 0);
           data1_ram_in  : in      std_logic_vector (15 downto 0);
           data2_ram_out : out     std_logic_vector (15 downto 0);
           data2_ram_in  : in      std_logic_vector (15 downto 0) );
```

```
end mux_data;
```

```
architecture arc_mux_data of mux_data is
```

```
    signal flag2in      : std_logic;
```

```
begin
```

```
    process (flag2in, data_dsp_in, data_cam_in)
```

```
    begin
```

```
        case flag2in is
```

```
            when '0' =>
```

```
                data1_ram_out <= data_dsp_in;
```

```
                data2_ram_out <= data_cam_in;
```

```
            when '1' =>
```

```
                data1_ram_out <= data_cam_in;
```

```
                data2_ram_out <= data_dsp_in;
```

```
            when others =>
```

```
                end case;
```

```
        end process;
```

```
    process (flag2in, data1_ram_in, data2_ram_in)
```

```
    begin
```

```
        case flag2in is
```

```
            when '0' => ram_im_r <= data1_ram_in;
```

```
            when '1' => ram_im_r <= data2_ram_in;
```

```
            when others =>
```

```
                end case;
```

```
        end process;
```

```
    process(rst, ck)
```

```
begin
  if rst = '1' then
    flag2in <= '0';
  elsif ck'event and ck = '1' then
    flag2in <= flag2;
  end if;
end process;

end arc_mux_data;
```

5.1.5. counter_addr.vhd

```
-- counter_addr.vhd

library ieee;
use ieee.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity counter_addr is
  port ( rst          : in          std_logic;
        end_im       : in          std_logic;
        rst_row      : in          std_logic;
        im_we        : in          std_logic;
        addr_cam     : out         std_logic_vector (17 downto 0) );
end counter_addr;

architecture arc_counter_addr of counter_addr is

  signal rst0       : std_logic;
  signal rst1       : std_logic;
  signal q0         : std_logic_vector (8 downto 0);
  signal q1         : std_logic_vector (8 downto 0);

begin

  process(rst0, im_we)
  begin
    if rst0 = '1' then
      q0 <= (others => '0');
    elsif (im_we'event and im_we = '0') then
      q0 <= q0 + '1';
    end if;
  end process;

  process(rst1, rst_row)
  begin
    if rst1 = '1' then
      q1 <= (others => '0');
    elsif (rst_row'event and rst_row = '1') then
      q1 <= q1 + '1';
    end if;
  end process;

end arc_counter_addr;
```

```

end if;

end process;

rst0 <= rst or end_im or rst_row;
rst1 <= rst or end_im;

addr_cam <= q1 & q0;

end arc_counter_addr;

```

5.1.6. serial.vhd

```

-- serial.vhd

library ieee;
use ieee.std_logic_1164.all;

entity serial is
    port ( rst           : in      std_logic;
           ck20          : in      std_logic;
           ser_rx        : in      std_logic;
           ser_tx        : out     std_logic;
           data_dsp_in   : in      std_logic_vector (7 downto 0);
           ser_r         : out     std_logic_vector (7 downto 0);
           addr_dsp      : in      std_logic_vector (2 downto 0);
           inttx         : out     std_logic;
           intrx         : out     std_logic;
           write_ser     : in      std_logic; -- write enable
           read_ser      : in      std_logic; -- read enable
           dms3          : in      std_logic );
end serial;

architecture arc_serial of serial is

    component Uart_top
    port (
        CK20    : in  std_logic; -- Input clock (20 MHz)
        MR      : in  std_logic;
        A       : in  std_logic_vector(2 downto 0); -- Address bus
        DIN     : in  std_logic_vector(7 downto 0); -- Data bus input
        DOUT    : out std_logic_vector(7 downto 0); -- Data but output
        ADSn    : in  std_logic; -- Address strobe
        -- CS    : in  std_logic; -- Chip Select
        RDn     : in  std_logic; -- Read
        WRn     : in  std_logic; -- Write
        -- DDIS  : out std_logic; -- Driver disable
        -- INTR  : out std_logic; -- Interrupt
        SIN     : in  std_logic; -- Receiver serial input
        RxRDYn  : out std_logic; -- Receiver ready
        SOUT    : out std_logic; -- Transmitter serial output
        TxRDYn  : out std_logic ); -- Transmitter ready
    end component;

```

```
begin
```

```
inst0 : Uart_top
  port map ( CK20=>CK20, MR=>RST,
            A(2 downto 0)=>addr_dsp(2 downto 0),
            DIN(7 downto 0)=>data_dsp_in(7 downto 0),
            DOUT(7 downto 0)=>ser_r(7 downto 0),
            ADSn=>dms3,
            RDn=>read_ser, WRn=>write_ser,
            SIN=>ser_rx, RxRDYn=>intrx, SOUT=>ser_tx, TxRDYn=>inttx );
```

```
end arc_serial;
```

5.1.7. uart_top.vhd

```
-- uart_top.vhd
```

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_Unsigned.all;
```

```
entity uart_top is
  port (
    CK20    : in  std_logic; -- Input clock (20 MHz)
    MR      : in  std_logic;
    A       : in  std_logic_vector(2 downto 0); -- Address bus
    DIN     : in  std_logic_vector(7 downto 0); -- Data bus input
    DOUT    : out std_logic_vector(7 downto 0); -- Data but output
    ADSn    : in  std_logic; -- Address strobe
    -- CS    : in  std_logic; -- Chip Select
    RDn     : in  std_logic; -- Read
    WRn     : in  std_logic; -- Write
    -- DDIS  : out std_logic; -- Driver disable
    -- INTR  : out std_logic; -- Interrupt
    SIN     : in  std_logic; -- Receiver serial input
    RxRDYn : out std_logic; -- Receiver ready
    SOUT    : out std_logic; -- Transmitter serial output
    TxRDYn : out std_logic ); -- Transmitter ready
end uart_top;
```

```
architecture uart_top_a of uart_top is
  component Interface
    port (CK20    : in  std_logic;
          Reset   : in  std_logic;
          Clk16X  : in  std_logic;
          A       : in  std_logic_vector(2 downto 0);
          DIN     : in  std_logic_vector(7 downto 0);
          DOUT    : out std_logic_vector(7 downto 0);
          ADSn    : in  std_logic;
          CS      : in  std_logic;
          RDn     : in  std_logic;
          WRn     : in  std_logic;
          -- DDIS  : out std_logic;
          -- INTR  : out std_logic;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 43/131
Filename: *ASTEDetailedDesign6*

```
RBR      : in  std_logic_vector(7 downto 0);
THR      : out std_logic_vector(7 downto 0);
MSR      : in  std_logic_vector(7 downto 0);
MCR      : out std_logic_vector(1 downto 0);
RbrRDn_re : out  std_logic;
ThrWRn_re : out  std_logic;
LsrRDn_re : inout std_logic;
MsrRDn_re : inout std_logic;
Databits  : out std_logic_vector(1 downto 0);
Stopbits  : out std_logic_vector(1 downto 0);
ParityEnable: out std_logic;
ParityEven : out std_logic;
ParityStick : out std_logic;
TxBreak   : out std_logic;
RxDY      : in  std_logic;
OverrunErr : in  std_logic;
ParityErr  : in  std_logic;
FrameErr   : in  std_logic;
BreakInt   : in  std_logic;
THRE      : in  std_logic;
TEMT      : in  std_logic
);
end component;

component Rxcver
port (CK20      : in  std_logic;
Reset         : in  std_logic;
Clk16X        : in  std_logic;
RBR           : out std_logic_vector(7 downto 0);
RbrRDn_re     : in  std_logic;
LsrRDn_re     : in  std_logic;
SIN           : in  std_logic;
Databits      : in  std_logic_vector(1 downto 0);
ParityEnable  : in  std_logic; -- 0= Parity Disabled; 1= Parity Enabled
ParityEven    : in  std_logic; -- 0= Odd Parity; 1= Even Parity
ParityStick   : in  std_logic; -- 0= Stick Disabled; 1= Stick Enabled
RxDY          : out std_logic; -- Receiver data ready to read
OverrunErr    : out std_logic; -- Receiver overrun error flag
ParityErr     : out std_logic; -- Receiver parity error flag
FrameErr      : out std_logic; -- Receiver framing error flag
BreakInt      : out std_logic  -- Receiver BREAK interrupt flag
);
end component;

component Txmitt
port (
Reset         : in  std_logic;
Clk16X        : in  std_logic;
THR           : in  std_logic_vector(7 downto 0); -- Transmitter Holding Reg
ThrWRn_re     : in  std_logic; -- pulse indicating rising of ThrWRn_r
SOUT          : out std_logic;
DataBits      : in  std_logic_vector(1 downto 0);
StopBits      : in  std_logic_vector(1 downto 0);
ParityEnable  : in  std_logic;
ParityEven    : in  std_logic;
ParityStick   : in  std_logic;
TxBreak       : in  std_logic;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 44/131
Filename: *ASTEDetailedDesign6*

```
    THRE      : out std_logic;
    TEMT      : out std_logic
);
end component;

component Counter
port (
    Reset      : in  std_logic;
    CK20       : in  std_logic;
    MCLK       : out std_logic );
end component;

signal MCLK      : std_logic; -- Master clock

signal CS        : std_logic;
signal RBR       : std_logic_vector(7 downto 0);
signal THR       : std_logic_vector(7 downto 0);
signal MSR       : std_logic_vector(7 downto 0);
signal MCR       : std_logic_vector(1 downto 0);

signal DataBits  : std_logic_vector(1 downto 0);
signal StopBits  : std_logic_vector(1 downto 0);
signal ParityEnable: std_logic;
signal ParityEven : std_logic;
signal ParityStick : std_logic;
signal TxBreak   : std_logic;

signal ThrWRn_re : std_logic;
signal RbrRDn_re : std_logic;
signal LsrRDn_re : std_logic;
signal MsrRDn_re : std_logic;

signal RxDY      : std_logic;
signal ParityErr : std_logic;
signal FrameErr  : std_logic;
signal OverrunErr : std_logic;
signal BreakInt  : std_logic;

signal THRE      : std_logic;
signal TEMT      : std_logic;

signal d         : std_logic_vector(7 downto 0);
signal c         : std_logic;
signal c1        : std_logic;
signal b         : std_logic;
signal b1        : std_logic;

begin

U1: Intface port map(
    CK20      => CK20,
    Reset     => MR,
    Clk16X    => MCLK,
--    Clk16X    => CK20,
    A(2 downto 0) => A(2 downto 0),
    DIN(7 downto 0) => d(7 downto 0),
    DOUT(7 downto 0) => DOUT(7 downto 0),
```

```

ADSn          => ADSn,
CS            => CS,
RDn          => RDn,
--   WRn      => WRn,
--   WRn      => b,
--   DDIS     => DDIS,
--   INTR     => INTR,
RBR(7 downto 0) => RBR(7 downto 0),
THR(7 downto 0) => THR(7 downto 0),
MSR(7 downto 0) => MSR(7 downto 0),
MCR(1 downto 0) => MCR(1 downto 0),
RbrRDn_re    => RbrRDn_re,
ThrWRn_re    => ThrWRn_re,
LsrRDn_re    => LsrRDn_re,
MsrRDn_re    => MsrRDn_re,
DataBits(1 downto 0) => DataBits(1 downto 0),
StopBits(1 downto 0) => StopBits(1 downto 0),
ParityEnable  => ParityEnable,
ParityEven    => ParityEven,
ParityStick   => ParityStick,
TxBreak       => TxBreak,
RxRDY        => RxRDY,
OverrunErr    => OverrunErr,
ParityErr     => ParityErr,
FrameErr      => FrameErr,
BreakInt      => BreakInt,
THRE          => THRE,
TEMT         => TEMT );

```

```

U2: Rxcver port map(CK20 => CK20,
Reset          => MR,
Clk16X        => MCLK,
RBR(7 downto 0) => RBR(7 downto 0),
RbrRDn_re    => RbrRDn_re,
LsrRDn_re    => LsrRDn_re,
SIN          => SIN,
DataBits(1 downto 0) => DataBits(1 downto 0),
ParityEnable  => ParityEnable,
ParityEven    => ParityEven,
ParityStick   => ParityStick,
RxRDY        => RxRDY,
OverrunErr    => OverrunErr,
ParityErr     => ParityErr,
FrameErr      => FrameErr,
BreakInt      => BreakInt );

```

```

U3: Txmitt port map(
Reset          => MR,
Clk16X        => MCLK,
THR(7 downto 0) => THR(7 downto 0),
ThrWRn_re    => ThrWRn_re,
DataBits(1 downto 0) => DataBits(1 downto 0),
StopBits(1 downto 0) => StopBits(1 downto 0),
ParityEnable  => ParityEnable,
ParityEven    => ParityEven,
ParityStick   => ParityStick,
TxBreak       => TxBreak,

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 46/131
Filename: ASTEDetailedDesign6

```
SOUT          => SOUT,  
THRE          => THRE,  
TEMT          => TEMT );
```

```
U4: Counter port map(  
  Reset       => MR,  
  CK20        => CK20,  
  MCLK        => MCLK );
```

```
-- TxRDYn <= not THRE;  
TxRDYn <= not TEMT;  
RxRDYn <= not RxRDY;
```

```
CS <= '1';  
MSR <= "00000000";
```

```
process(WRn, MR)  
begin  
  if MR = '1' then  
    d <= (others=>'0');  
  elsif (WRn'event and WRn = '1') then  
    d <= DIN;  
  end if;  
end process;
```

```
process(MCLK, WRn, MR)  
begin  
  if (WRn = '0' or MR = '1') then  
    c <= '0';  
    b <= '0';  
  elsif (MCLK'event and MCLK = '0') then  
    c <= c1;  
    b <= b1;  
  end if;  
end process;
```

```
c1 <= (not c and not b) or (not c and b) or (c and b);  
b1 <= (c and not b) or (not c and b) or (c and b);
```

```
end uart_top_a;
```

5.1.8. interface.vhd

```
-- Interface.vhd
```

```
library IEEE;  
use IEEE.Std_Logic_1164.all;  
use IEEE.std_logic_unsigned.all;
```

```
entity Interface is  
  port (CK20      : in    std_logic;  
        Reset    : in    std_logic;  
        Clk16X   : in    std_logic;  
        A        : in    std_logic_vector(2 downto 0);  
        DIN      : in    std_logic_vector(7 downto 0);  
        DOUT     : out   std_logic_vector(7 downto 0);
```

```

ADSn      : in      std_logic;
CS        : in      std_logic;
RDn       : in      std_logic;
WRn       : in      std_logic;
--  DDIS   : out     std_logic;
--  INTR   : out     std_logic;
RBR       : in      std_logic_vector(7 downto 0);
THR       : out     std_logic_vector(7 downto 0);
MSR       : in      std_logic_vector(7 downto 0);
MCR       : out     std_logic_vector(1 downto 0);
RbrRDn_re : out     std_logic;
ThrWRn_re : out     std_logic;
LsrRDn_re : inout   std_logic;
MsrRDn_re : inout   std_logic;
Databits  : out     std_logic_vector(1 downto 0);
Stopbits  : out     std_logic_vector(1 downto 0);
ParityEnable: out   std_logic;
ParityEven : out   std_logic;
ParityStick : out   std_logic;
TxBreak   : out     std_logic;
RxRDY     : in      std_logic;
OverrunErr : in     std_logic;
ParityErr  : in     std_logic;
FrameErr   : in     std_logic;
BreakInt   : in     std_logic;
THRE      : in     std_logic;
TEMT      : in     std_logic );

```

end Intface;

architecture Intface_a of Intface is

```

signal ADDR_s      : std_logic_vector(2 downto 0);
signal CS_r        : std_logic;

signal WRn_cs      : std_logic;
signal RDn_cs      : std_logic;

signal LSR         : std_logic_vector(6 downto 0);
-- signal LCR       : std_logic_vector(6 downto 0);
signal IIR         : std_logic_vector(3 downto 0);
signal IER         : std_logic_vector(3 downto 0);

signal ThrWRn_r    : std_logic;
signal RbrRDn_r    : std_logic;
signal LsrRDn_r    : std_logic;
signal MsrRDn_r    : std_logic;
signal IirRDn_r    : std_logic;
signal IirRDn_re   : std_logic;

signal ThrWRn1_r   : std_logic;
signal ThrWRn2_r   : std_logic;
signal RbrRDn1_r   : std_logic;
signal RbrRDn2_r   : std_logic;
signal LsrRDn1_r   : std_logic;
signal LsrRDn2_r   : std_logic;
signal MsrRDn1_r   : std_logic;

```

```

signal MsrRDn2_r : std_logic;
signal IirRDn1_r : std_logic;
signal IirRDn2_r : std_logic;

signal RxRDY_Int   : std_logic;
signal THRE_Int    : std_logic;
signal DataErr_Int : std_logic;
signal Modem_Int   : std_logic;

signal DataErr      : std_logic;
signal ModemStat    : std_logic;

type state_typ is (idle, int0, int1, int2, int3);
signal Int_State : state_typ;

constant A_RBR : std_logic_vector(2 downto 0) := "000";
constant A_THR : std_logic_vector(2 downto 0) := "000";
constant A_IER : std_logic_vector(2 downto 0) := "001";
constant A_IIR : std_logic_vector(2 downto 0) := "010";
-- constant A_LCR : std_logic_vector(2 downto 0) := "011";
constant A_MCR : std_logic_vector(2 downto 0) := "100";
constant A_LSR : std_logic_vector(2 downto 0) := "101";
constant A_MSR : std_logic_vector(2 downto 0) := "110";

begin

Addr_Latch_Proc: process(Reset, ADSn, A)
begin
    if Reset='1' then
        ADDR_s <= (others=>'0');
    elsif ADSn='0' then
        ADDR_s <= A;
    end if;
end process Addr_Latch_Proc;

Chip_Select_Latch_Proc: process(Reset, CS, ADSn)
begin
    if Reset='1' then
        CS_r <= '0';
    elsif ADSn='0' then
        CS_r <= CS;
    end if;
end process Chip_Select_Latch_Proc;

WRn_cs <= WRn when (CS_r='1') else '1';
RDn_cs <= RDn when (CS_r='1') else '1';

ThrWRn_r <= '1' when (Reset='1') else WRn_cs when (ADDR_s=A_THR) else '1';
RbrRDn_r <= '1' when (Reset='1') else RDn_cs when (ADDR_s=A_RBR) else '1';
LsrRDn_r <= '1' when (Reset='1') else RDn_cs when (ADDR_s=A_LSR) else '1';
MsrRDn_r <= '1' when (Reset='1') else RDn_cs when (ADDR_s=A_MSR) else '1';
IirRDn_r <= '1' when (Reset='1') else RDn_cs when (ADDR_s=A_IIR) else '1';

Delay_Signals_Proc: process(CK20, Reset)
begin
    if (Reset='1') then
        RbrRDn1_r <= '1';
    end if;
end process Delay_Signals_Proc;

```

```

    RbrRDn2_r <= '1';
    elsif rising_edge(CK20) then
        RbrRDn1_r <= RbrRDn_r;
        RbrRDn2_r <= RbrRDn1_r;
    end if;
end process Delay_Signals_Proc;

Delay_Signals_Proc1: process(Clk16X, Reset)
-- Delay_Signals_Proc: process(CK20, Reset)
begin
    if (Reset='1') then
        ThrWRn1_r <= '1';
        ThrWRn2_r <= '1';
--        RbrRDn1_r <= '1';
--        RbrRDn2_r <= '1';
        LsrRDn1_r <= '1';
        LsrRDn2_r <= '1';
        MsrRDn1_r <= '1';
        MsrRDn2_r <= '1';
        IirRDn1_r <= '1';
        IirRDn2_r <= '1';
    elsif rising_edge(Clk16X) then
--        elsif rising_edge(CK20) then
        ThrWRn1_r <= ThrWRn_r;
        ThrWRn2_r <= ThrWRn1_r;
--        RbrRDn1_r <= RbrRDn_r;
--        RbrRDn2_r <= RbrRDn1_r;
        LsrRDn1_r <= LsrRDn_r;
        LsrRDn2_r <= LsrRDn1_r;
        MsrRDn1_r <= MsrRDn_r;
        MsrRDn2_r <= MsrRDn1_r;
        IirRDn1_r <= IirRDn_r;
        IirRDn2_r <= IirRDn1_r;
    end if;
end process Delay_Signals_Proc1;

ThrWRn_re <= ThrWRn1_r and (not ThrWRn2_r);
RbrRDn_re <= RbrRDn1_r and (not RbrRDn2_r);
LsrRDn_re <= LsrRDn1_r and (not LsrRDn2_r);
MsrRDn_re <= MsrRDn1_r and (not MsrRDn2_r);
IirRDn_re <= IirRDn1_r and (not IirRDn2_r);

Rd_Register_Proc:
process (Reset, RDn_cs, ADDR_s, RBR, IIR, LSR, MSR)
begin
    if (Reset='1') then
        DOUT <= (others => '1');
    elsif (RDn_cs ='0') then
        case ADDR_s is
            when A_RBR =>
                DOUT <= RBR;
            when A_IIR =>
                DOUT <= "0000" & IIR;
            when A_LSR =>
                DOUT <= '0' & LSR;
            when A_MSR =>

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 50/131
Filename: ASTEDetailedDesign6

```
DOUT <= MSR;
when others =>
    DOUT <= (others => '1');
end case;
else
    DOUT <= (others=>'1');
end if;
end process Rd_Register_Proc;

Wr_Register_Proc:
process (Reset, WRn_cs)
begin
    if (Reset='1') then
        THR <= (Others=>'0');
--        IER <= (Others=>'0');
--        LCR <= (Others=>'0');
        MCR <= (Others=>'0');
    elsif rising_edge(WRn_cs) then
        case ADDR_s is
            when A_THR =>
                THR <= DIN;
--            when A_IER =>
--                IER <= DIN(3 downto 0);
--            when A_LCR =>
--                LCR <= DIN(6 downto 0);
            when A_MCR =>
                MCR <= DIN(1 downto 0);
            when others =>
                end case;
        end if;
    end process Wr_Register_Proc;

IER <= "0011";

-- Databits : "00"=5-bit, "01"=6-bit, "10"=7-bit, "11"=8-bit
-- Databits <= LCR(1 downto 0);
Databits <= "11";

-- Stopbits : "00"=1-bit, "01"=1.5-bit(5-bit data), "10"=2-bit(6,7,8-bit data)
-- Stopbits <= "00" when (LCR(2)='0') else
--             "01" when (LCR(2)='1') and (LCR(1 downto 0)="00") else
--             "10";
Stopbits <= "00";

-- ParityEnable : '1'=Parity Bit Enable, '0'=Parity Bit Disable
-- ParityEnable <= LCR(3);
ParityEnable <= '0';    -- *****

-- ParityEven : '0'=Even Parity Selected, '1'=Odd Parity Selected
-- ParityEven <= LCR(4);
ParityEven <= '0';

-- ParityStick : '0'=Stick Parity Disable, '1'=Stick Parity Enable
--ParityStick <= LCR(5);
ParityStick <= '0';

-- TxBreak : '0'=Disable BREAK assertion, '1'=Assert BREAK
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 51/131
Filename: ASTEDetailedDesign6

```
-- TxBreak <= LCR(6);  
TxBreak <= '0';  
  
LSR <= TEMT & THRE & BreakInt & FrameErr & ParityErr & OverrunErr & RxRDY;  
  
-- Int is the common interrupt line for all internal UART events  
-- INTR <= RxRDY_Int or THRE_Int or DataErr_Int or Modem_Int;  
  
-- Receiving Data Error Flags including Overrun, Parity, Framing and Break  
DataErr <= OverrunErr or ParityErr or FrameErr or BreakInt;  
  
-- Whenever bit0,1,2,or3 is set to '1', a Modem Status Interrupt is generated  
ModemStat <= MSR(0) or MSR(1) or MSR(2) or MSR(3);  
  
-- Int_Arbit_Proc: process(Reset, Clk16X)  
Int_Arbit_Proc: process(Reset, CK20)  
begin  
  if (Reset='1') then  
    Int_State <= idle;  
  
--  elsif rising_edge(Clk16X) then  
  elsif rising_edge(CK20) then  
    case Int_State is  
      when idle =>  
  
        if (IER(2)='1') and (DataErr='1') then  
          Int_State <= int0;  
        elsif (IER(0)='1') and (RxRDY='1') then  
          Int_State <= int1;  
        elsif (IER(1)='1') and (THRE='1') then  
          Int_State <= int2;  
        elsif (IER(3)='1') and (ModemStat='1') then  
          Int_State <= int3;  
        end if;  
  
      when int0 =>  
        if (LsrRDn_re='1') then  
          -- Clear Receiver Line Status Interrupt after LSR read  
          Int_State <= idle;  
        end if;  
  
      when int1 =>  
        if (RxRDY='0') then  
          -- Clear Received Data Available Interrupt after RBR read(RxRDY=0)  
          Int_State <= idle;  
        end if;  
  
      when int2 =>  
        if (IirRDn_re='1') or (THRE='0') then  
          -- Clear THR Empty Interrupt after IIR read or THR write(THRE=0)  
          Int_State <= idle;  
        end if;  
  
      when int3 =>  
        if (MsrRDn_re='1') then  
          -- Clear MODEM Status Interrupt after MSR read  
          Int_State <= idle;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 52/131
Filename: *ASTEDetailedDesign6*

```
        end if;
        when others =>
            Int_State <= idle;
        end case;
    end if;
end process Int_Arbit_Proc;

-- Set Receiver Line Status Interrupt
DataErr_Int <= '1' when Int_State = int0 else '0';

-- Set Received Data Available Interrupt
RxRDY_Int <= '1' when Int_State = int1 else '0';

-- Set THR Empty Interrupt
THRE_Int <= '1' when Int_State = int2 else '0';

-- Set MODEM Status Interrupt
Modem_Int <= '1' when Int_State = int3 else '0';

-- Update IIR
IIR <= "0110" when Int_State = int0 else -- 1st Priority Interrupt
      "0100" when Int_State = int1 else -- 2nd Priority Interrupt
      "0010" when Int_State = int2 else -- 3rd Priority Interrupt
      "0000" when Int_State = int3 else -- 4th Priority Interrupt
      "0001";                          -- No Interrupt Pending

-- Driver Disable goes low whenever the CPU is reading data from the UART
-- DDIS <= RDn_cs;

end Interface_a;
```

5.1.9. rxcver.vhd

```
-- Rxcver.vhd

library IEEE;
use IEEE.Std_Logic_1164.all;
use IEEE.Std_Logic_Unsigned.all;

entity Rxcver is
    port (CK20      : in  std_logic;
          Reset     : in  std_logic;
          Clk16X    : in  std_logic;
          RBR       : out std_logic_vector(7 downto 0);
          RbrRDn_re : in  std_logic;
          LsrRDn_re : in  std_logic;
          SIN       : in  std_logic;
          Databits  : in  std_logic_vector(1 downto 0);
          ParityEnable: in  std_logic;
          ParityEven  : in  std_logic;
          ParityStick : in  std_logic;
          RxRDY      : out std_logic;
          OverrunErr : out std_logic);
end entity Rxcver;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 53/131
Filename: *ASTEDetailedDesign6*

```
    ParityErr      : out std_logic;
    FrameErr      : out std_logic;
    BreakInt      : out std_logic
);
end Rxcver;

architecture Rxcver_a of Rxcver is

    signal NumDataBitReceived_r : std_logic_vector(3 downto 0);

    signal RSR          : std_logic_vector(7 downto 0);
    signal RxPrtyErr    : std_logic;
    signal RxFrMErr     : std_logic;
    signal RxIdle_r     : std_logic;
    signal RbrDataRDY   : std_logic;
    signal CNT_r        : std_logic_vector(3 downto 0);

    signal Hunt_r       : boolean;
    signal HuntOne_r    : std_logic;

    signal SIN1_r       : std_logic;
    signal RxFrMErr1_r  : std_logic;
    signal RxIdle1_r    : std_logic;

    signal OverrunErr_r : std_logic;
    signal ParityErr_r  : std_logic;
    signal FrameErr_r   : std_logic;
    signal BreakInt_r   : std_logic;

    signal SampledOnce : std_logic;

    signal RxClkEn      : std_logic;

    signal RBR_r        : std_logic_vector(7 downto 0);

    type state_typ is (idle, shift, parity, stop);
    signal Rx_State : state_typ;

    signal Clk10        : std_logic;

begin

    RxCLK_Proc: process (Reset, Clk16X)
    begin
        if (Reset='1') then
            RxClkEn <= '0';
        elsif rising_edge(Clk16X) then
            if (CNT_r="0110") then
                RxClkEn <= '1';
            else
                RxClkEn <= '0';
            end if;
        end if;
    end process RxCLK_Proc;

    CNT_Proc: process (Reset, Clk16X)
    begin
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 54/131
Filename: ASTEDetailedDesign6

```
if (Reset='1') then
  CNT_r <= (others => '0');
elsif rising_edge(Clk16X) then
  if (Rx_State /= idle) or (Hunt_r) then
    CNT_r <= CNT_r + 1;
  elsif (SampledOnce='1') then
    CNT_r <= "0010";
  else
    CNT_r <= (others => '0');
  end if;
end if;
end process CNT_Proc;

Hunt_r_Proc: process (Reset, Clk16X)
begin
  if (Reset='1') then
    Hunt_r <= FALSE;
  elsif rising_edge(Clk16X) then
    if (Rx_State=idle) and (SIN='0') and (SIN1_r='1') then
      Hunt_r <= TRUE;
    elsif (SampledOnce='1') and (SIN='0') then
      Hunt_r <= TRUE;
    elsif (RxIdle_r='0') or (SIN='1') then
      Hunt_r <= FALSE;
    end if;
  end if;
end process Hunt_r_Proc;

HuntOne_r_Proc: process(Clk16X, Reset)
begin
  if (Reset='1') then
    HuntOne_r <= '0';
  elsif rising_edge(Clk16X) then
    if (Hunt_r) then
      HuntOne_r <= '0';
    elsif (RxIdle_r='0') and (CNT_r(3)='1') and (SIN='1') then
      HuntOne_r <= '1';
    end if;
  end if;
end process HuntOne_r_Proc;

-- RbrDataRDY_Proc: process(Clk16X, Reset)
RbrDataRDY_Proc: process(Clk10, Reset)
begin
  if (Reset='1') then
    RbrDataRDY <= '0';
  elsif rising_edge(Clk10) then
    if (RxIdle_r='1') and (RxIdle1_r='0') then
      RbrDataRDY <= '1';
    elsif (RbrRDn_re='1') then
      RbrDataRDY <= '0';
    end if;
  end if;
end process RbrDataRDY_Proc;

SampledOnce_Proc: process (Reset, Clk16X)
begin
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 55/131
Filename: *ASTEDetailedDesign6*

```
if (Reset='1') then
    SampledOnce <= '0';
elsif rising_edge(Clk16X) then
    if (RxFrmErr='1') and (RxFrmErr1_r='0') and
        (SIN='0') and (HuntOne_r='1') then
        SampledOnce <= '1';
    else
        SampledOnce <= '0';
    end if;
end if;
end process SampledOnce_Proc;

RxIdle_Proc: process(Reset, Clk16X)
begin
    if (Reset='1') then
        RxIdle_r <= '1';
    elsif rising_edge(Clk16X) then
        if (Rx_State=idle) then
            RxIdle_r <= '1';
        elsif (CNT_r(3)='1') then
            RxIdle_r <= '0';
        end if;
    end if;
end process RxIdle_Proc;

Shift_data_Proc: process(Clk16X, Reset)
begin
    if (Reset='1') then
        RSR <= (others=>'0');
        NumDataBitReceived_r <= (others=>'0');
        RxPrtyErr <= '1';
        RxFrmErr <= '0';
        Rx_State <= idle;
    elsif rising_edge(Clk16X) then
        case Rx_State is
            when idle =>
                if (RxIdle_r='1') and (SIN='0') and (RxClkEn='1') then
                    RSR <= (others=>'0');
                    NumDataBitReceived_r <= (others=>'0');
                    RxPrtyErr <= not ParityEven;
                    RxFrmErr <= '0';
                    Rx_State <= shift;
                end if;
            when shift =>
                if (RxClkEn='1') then
                    RSR <= SIN & RSR(7 downto 1);
                    RxPrtyErr <= RxPrtyErr xor SIN;
                    NumDataBitReceived_r <= NumDataBitReceived_r + 1;
                    if ((DataBits="00" and NumDataBitReceived_r=4) or
                        (DataBits="01" and NumDataBitReceived_r=5) or
                        (DataBits="10" and NumDataBitReceived_r=6) or
                        (DataBits="11" and NumDataBitReceived_r=7)) then
                        if (ParityEnable='0') then
                            Rx_State <= stop;
                        else
                            Rx_State <= parity;
                        end if;
                    end if;
                end if;
            end case;
    end if;
end process Shift_data_Proc;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 56/131
Filename: ASTEDetailedDesign6

```
        end if;
    end if;
    when parity =>
        if (RxClkEn='1') then
            if (ParityStick='0') then
                RxPrtyErr <= RxPrtyErr xor SIN;
            else
                if (ParityEven='0') then
                    RxPrtyErr <= not SIN;
                else
                    RxPrtyErr <= SIN;
                end if;
            end if;
            Rx_State <= stop;
        end if;
    when stop =>
        if (RxClkEn='1') then
            RxFrmErr <= not SIN;
            Rx_State <= idle;
        end if;
    when others =>
        if (RxClkEn='1') then
            Rx_State <= idle;
        end if;
    end case;
end if;
end process Shift_Data_Proc;

RBR_Proc: process(Clk16X, Reset)
begin
    if (Reset='1') then
        RBR_r <= (others=>'0');
    elsif rising_edge(Clk16X) then
        if (RxIdle_r='1') and (RxIdle1_r='0') then
            case DataBits is
                when "00" => -- 5-bit data
                    RBR_r <= "000" & RSR(7 downto 3);
                when "01" => -- 6-bit data
                    RBR_r <= "00" & RSR(7 downto 2);
                when "10" => -- 7-bit data
                    RBR_r <= '0' & RSR(7 downto 1);
                when others => -- 8-bit data
                    RBR_r <= RSR;
            end case;
        end if;
    end if;
end process RBR_Proc;

RBR <= RBR_r;

Delay_Signals_Proc: process(Clk16X, Reset)
begin
    if (Reset='1') then
        SIN1_r <= '0';
        RxFrmErr1_r <= '1';
        RxIdle1_r <= '1';
    elsif rising_edge(Clk16X) then
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 57/131
Filename: *ASTEDetailedDesign6*

```
SIN1_r <= SIN;
RxFrmErr1_r <= RxFrmErr;
RxIdle1_r <= RxIdle_r;
end if;
end process Delay_Signals_Proc;

Error_Flags_Proc: process(Clk16X, Reset)
begin
  if (Reset='1') then
    OverrunErr_r <= '0';
    ParityErr_r <= '0';
    FrameErr_r <= '0';
    BreakInt_r <= '0';
  elsif rising_edge(Clk16X) then
    if (RxIdle_r='1') and (RxIdle1_r='0') then
      OverrunErr_r <= RbrDataRDY;
      ParityErr_r <= (ParityErr_r or RxPrtyErr) and ParityEnable;
      FrameErr_r <= FrameErr_r or RxFrmErr;
      BreakInt_r <= BreakInt_r or (not HuntOne_r);
    elsif (LsrRDn_re='1') then
      ParityErr_r <= '0';
      FrameErr_r <= '0';
      OverrunErr_r <= '0';
      BreakInt_r <= '0';
    end if;
  end if;
end process Error_Flags_Proc;
```

```
RxRDY <= RbrDataRDY;
```

```
OverrunErr <= OverrunErr_r;
```

```
ParityErr <= ParityErr_r;
```

```
FrameErr <= FrameErr_r;
```

```
BreakInt <= BreakInt_r;
```

-- *****

```
process(Ck20, Reset)
begin
  if Reset = '1' then
    Clk10 <= '0';
  elsif (Ck20'event and Ck20 = '1') then
    Clk10 <= not Clk10;
  end if;
end process;
```

```
end Rxcver_a;
```

5.1.10. txmitt.vhd

-- Txmitt.vhd



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 58/131
Filename: ASTEDetailedDesign6

```
library IEEE;
use IEEE.Std_Logic_1164.all;
use IEEE.Std_Logic_Unsigned.all;

entity Txmitt is
  port (
    Reset      : in  std_logic;
    Clk16X     : in  std_logic;
    THR        : in  std_logic_vector(7 downto 0);
    ThrWRn_re  : in  std_logic;
    SOUT       : out std_logic;
    DataBits   : in  std_logic_vector(1 downto 0);
    StopBits   : in  std_logic_vector(1 downto 0);
    ParityEnable: in  std_logic;
    ParityEven  : in  std_logic;
    ParityStick : in  std_logic;
    TxBreak    : in  std_logic;
    THRE       : out std_logic;
    TEMT       : out std_logic
  );
end Txmitt;

architecture Txmitt_a of Txmitt is

  signal TxOutput      : std_logic;
  signal TSR           : std_logic_vector(7 downto 0);
  signal TxParity_r    : std_logic;
  signal ThrEmpty      : std_logic;
  signal TsrEmpty      : std_logic;

  signal TxInStartState : std_logic;
  signal TxInShiftState : std_logic;
  signal TxInStopState  : std_logic;

  signal TxInStartState1 : std_logic;
  signal TxInShiftState1 : std_logic;
  signal TxInStopState1  : std_logic;

  signal TxClkEnA      : std_logic;
  signal TxClkEnB      : std_logic;

  signal TxCNT_r       : std_logic_vector(2 downto 0);
  signal Count_vr      : std_logic_vector(3 downto 0);

  type state_typ is (start, shift, parity, stop_1bit, stop_2bit, stop_halfbit);
  signal Tx_State      : state_typ;

begin

  Shift_Data_Proc: process(Reset, CLK16X)
  begin
    if (Reset='1') then
      TxCNT_r <= (others=>'0');
      TSR <= (others=>'0');
      TxOutput <= '1';
      TxParity_r <= '1';
    end if;
  end process;
end Txmitt_a;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 59/131
Filename: ASTEDetailedDesign6

```
Tx_State <= start;
--
elsif rising_edge(Clk16X) then
elsif (Clk16X'event and Clk16X = '0') then
case Tx_State is
when start =>
    if (ThrEmpty='0') and (TxClkEnA='1') then
        TSR <= THR;
        TxParity_r <= not ParityEven;
        TxOutput <= '0';
        TxCNT_r <= (others=>'0');
        Tx_State <= shift;
    else
        TxOutput <= '1';
    end if;
when shift =>
    if (TxClkEnA='1') then
        TSR <= '0' & TSR(7 downto 1);
        TxParity_r <= TxParity_r xor TSR(0);
        TxOutput <= TSR(0);
        TxCNT_r <= TxCNT_r + '1';
        if ((Databits="00" and TxCNT_r = "100") or
            (Databits="01" and TxCNT_r = "101") or
            (Databits="10" and TxCNT_r = "110") or
            (Databits="11" and TxCNT_r = "111")) then
            if (ParityEnable='0') then
                Tx_State <= stop_1bit;
            else
                Tx_State <= parity;
            end if;
        end if;
    end if;
when parity =>
    if (TxClkEnA='1') then
        if (ParityStick='0') then
            TxOutput <= TxParity_r;
        else
            TxOutput <= not ParityEven;
        end if;
        Tx_State <= stop_1bit;
    end if;
when stop_1bit =>
    if (TxClkEnA='1') then
        TxOutput <= '1';
        if (StopBits="00") then
            Tx_State <= start;
        elsif (StopBits="01") then
            Tx_State <= stop_halfbit;
        else
            Tx_State <= stop_2bit;
        end if;
    end if;
when stop_2bit =>
    if (TxClkEnA='1') then
        TxOutput <= '1';
        Tx_State <= start;
    end if;
when stop_halfbit =>
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 60/131
Filename: *ASTEDetailedDesign6*

```
        if (TxClkEnB='1') then
            TxOutput <= '1';
            Tx_State <= start;
        end if;
    when others =>
        Tx_State <= Start;
    end case;
end if;
end process Shift_Data_Proc;

TsrEmpty_Proc: process(Clk16X, Reset)
begin
    if (Reset='1') then
        TsrEmpty <= '1';
    elsif rising_edge(Clk16X) then
        if (TxInStopState='1') and (TxInStopState1='0') then
            TsrEmpty <= '1';
        elsif (TxInShiftState='1') and (TxInShiftState1='0') then
            TsrEmpty <= '0';
        end if;
    end if;
end process TsrEmpty_Proc;

ThrEmpty_Proc: process(Clk16X, Reset)
begin
    if (Reset='1') then
        ThrEmpty <= '1';
    elsif rising_edge(Clk16X) then
        if (ThrWRn_re='1') then
            ThrEmpty <= '0';
        elsif (TxInShiftState='1') and (TxInShiftState1='0') then
            ThrEmpty <= '1';
        end if;
    end if;
end process ThrEmpty_Proc;

Delay_Signals_Proc: process(Clk16X, Reset)
begin
    if (Reset='1') then
        TxInStartState1 <= '1';
        TxInShiftState1 <= '1';
        TxInStopState1 <= '1';
    elsif rising_edge(Clk16X) then
        TxInStartState1 <= TxInStartState;
        TxInShiftState1 <= TxInShiftState;
        TxInStopState1 <= TxInStopState;
    end if;
end process Delay_Signals_Proc;

TxInShiftState_Proc: process(Clk16X, Reset)
begin
    if (Reset='1') then
        TxInShiftState <= '0';
    elsif rising_edge(Clk16X) then
        if (Tx_State=shift) then
            TxInShiftState <= '1';
        else
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 61/131
Filename: ASTEDetailedDesign6

```
TxInShiftState <= '0';
end if;
end if;
end process TxInShiftState_Proc;

TxInStopState_Proc: process(Clk16X, Reset)
begin
  if (Reset='1') then
    TxInStopState <= '0';
  elsif rising_edge(Clk16X) then
    if (Tx_State=stop_1bit) then
      TxInStopState <= '1';
    else
      TxInStopState <= '0';
    end if;
  end if;
end process TxInStopState_Proc;

TxInStartState_Proc: process(Clk16X, Reset)
begin
  if (Reset='1') then
    TxInStartState <= '0';
  elsif rising_edge(Clk16X) then
    if (Tx_State=start) then
      TxInStartState <= '1';
    else
      TxInStartState <= '0';
    end if;
  end if;
end process TxInStartState_Proc;

TxCLK_Proc: process(Reset, Clk16X)
begin
  if (Reset='1') then
    Count_vr <= (others => '1');
    TxClkEnA <= '0';
    TxClkEnB <= '0';
  elsif rising_edge(Clk16X) then

    if (Count_vr="0000") then
      TxClkEnA <= '1';
    else
      TxClkEnA <= '0';
    end if;

    if (Count_vr="1000") then
      TxClkEnB <= '1';
    else
      TxClkEnB <= '0';
    end if;

    if (TxInStartState='1') and (TxInStartState1='0') then
      Count_vr <= "0011";
    else
      Count_vr <= Count_vr + '1';
    end if;
  end if;
end process TxCLK_Proc;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 62/131
Filename: *ASTEDetailedDesign6*

```
end if;
end process TxCLK_Proc;

THRE <= ThrEmpty;

TEMT <= '1' when (ThrEmpty='1') and (TsrEmpty='1') else '0';

SOUT <= '0' when (TxBreak='1') else TxOutput;

end Txmitt_a;
```

5.1.11. counter.vhd

```
-- Counter.vhd

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
-- use IEEE.std_logic_unsigned.all;

entity Counter is
port (
    Reset      : in  std_logic;
    Ck20       : in  std_logic;
    MCLK       : out std_logic );
end Counter;

architecture Counter_a of Counter is

constant count : integer range 0 to 127 := 65;

begin

process(Ck20, Reset)
variable cnt : integer range 0 to count-1 := count-1;
begin
    if Reset = '1' then
        MCLK <= '0';
        cnt := count-1;
    elsif (Ck20'event and Ck20 = '1') then
        if cnt = 0 then
            MCLK <= '1';
            cnt := count-1;
        else
            MCLK <= '0';
            cnt := cnt-1;
        end if;
    end if;
end process;

end Counter_a;
```

5.1.12. timer_data.vhd

```
-- timer_data.vhd
```

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity timer_data is
    port ( rst           : in      std_logic;
          ck20          : in      std_logic;
          sync          : in      std_logic;
          addr_dsp      : in      std_logic_vector (0 downto 0);
          timer         : out     std_logic_vector (15 downto 0) );
end timer_data;

architecture arc_timer_data of timer_data is

    signal ckdiv        : std_logic;
    signal sync_count   : std_logic;
    signal q             : std_logic_vector (31 downto 0);
    signal sel          : std_logic;

    constant count      : integer range 0 to 65535 := 20000;

begin

    process(ck20, rst)
        variable cnt : integer range 0 to count-1 := count-1;
    begin
        if rst = '1' then
            ckdiv <= '0';
            cnt := count-1;
        elsif (ck20'event and ck20 = '1') then
            if cnt = 0 then
                ckdiv <= '1';
                cnt := count-1;
            else
                ckdiv <= '0';
                cnt := cnt-1;
            end if;
        end if;
    end process;

    sync_count <= rst or sync;

    process(sync_count, ckdiv)
    begin
        if (sync_count = '1') then
            q <= (others => '0');
        elsif (ckdiv'event and ckdiv = '1') then
            q <= q + '1';
        end if;
    end process;

    process(sel, q)
    begin
        case sel is
            when '0' => timer <= q(15 downto 0);
        end case;
    end process;
end arc_timer_data;
```

```
when '1' => timer <= q(31 downto 16);
when others => timer <= (others => '0');
end case;
end process;

sel <= addr_dsp(0);

end arc_timer_data;
```

5.1.13. hk.vhd

```
-- hk.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity hk is
    port ( stx           : out  std_logic;
          strb          : out  std_logic;

          rst           : in   std_logic;
          hk_dout1      : in   std_logic;
--          sstrb1       : in   std_logic;
          hk_dout2      : in   std_logic;
--          sstrb2       : in   std_logic;
          hk_din        : out  std_logic;
          cs            : out  std_logic;
          ck            : in   std_logic; -- 20MHz
          sclk          : out  std_logic;
          hk_data       : out  std_logic_vector (15 downto 0);
          addr_dsp      : in   std_logic_vector (3 downto 0);
          dms3          : in   std_logic;
          irq0          : out  std_logic );
end hk;

architecture arc_hk of hk is

    component hk_adc
        port ( dout       : in   std_logic;
              sstrb      : in   std_logic;
              sclk       : in   std_logic;
              data       : out  std_logic_vector (11 downto 0);
              we         : out  std_logic );
    end component;

    component reg_sram
        port ( data1      : in   std_logic_vector (11 downto 0);
              data2      : in   std_logic_vector (11 downto 0);
              q           : out  std_logic_vector (15 downto 0);
              we1        : in   std_logic;
              we2        : in   std_logic;
              address    : in   std_logic_vector (3 downto 0);
```

```

        addrw          : in      std_logic_vector (2 downto 0);
        dms3           : in      std_logic );
end component;

signal  data1          : std_logic_vector (11 downto 0);
signal  data2          : std_logic_vector (11 downto 0);
signal  q4             : std_logic_vector (7 downto 0);
signal  addrw          : std_logic_vector (2 downto 0);

signal  count         : std_logic_vector (19 downto 0);
signal  ckser         : std_logic;
signal  sclk1         : std_logic;
signal  we1           : std_logic;
signal  we2           : std_logic;
signal  abil0         : std_logic;
signal  abil1         : std_logic;

-- loop

-- signal  hk_din      : std_logic;
signal  hk_din1       : std_logic;
-- signal  sstrb       : std_logic;
-- signal  hk_dout1    : std_logic;
-- signal  hk_dout2    : std_logic;
signal  sstrb1        : std_logic;
signal  sstrb2        : std_logic;
signal  st            : std_logic;
signal  st1           : std_logic;
-- signal  addr_dsp    : std_logic_vector (3 downto 0);

begin

inst0 : hk_adc
    port map ( dout=>hk_dout1, sstrb=>sstrb1, sclk=>sclk1,
               data(11 downto 0)=>data1(11 downto 0), we=>we1 );

inst1 : hk_adc
    port map ( dout=>hk_dout2, sstrb=>sstrb2, sclk=>sclk1,
               data(11 downto 0)=>data2(11 downto 0), we=>we2 );

inst2 : reg_sram
    port map ( data1(11 downto 0)=>data1(11 downto 0),
               data2(11 downto 0)=>data2(11 downto 0),
               q(15 downto 0)=>hk_data(15 downto 0),
               we1=>we1, we2=>we2,
               address(3 downto 0)=>addr_dsp(3 downto 0),
               addrw(2 downto 0)=>addrw(2 downto 0),
               dms3=>dms3 );

process (ck, rst)
begin
    if rst = '1' then
        count <= (others => '0');
    elsif (ck'event and ck = '1') then
        count <= count + '1';
    end if;
end process;

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 66/131
Filename: ASTEDetailedDesign6

```
process (ckser, count(9))
begin
  if (ckser'event and ckser = '1') then
    if (count(9) = '0') then
      q4 <= '1' & count(10) & count(12) & count(11) & "1111" ;
    else
      q4 <= q4(6 downto 0) & '0';
    end if;
  end if;
end process;

process (ckser, count(9))
begin
  if (count(9) = '0') then
    hk_din1 <= '0';
  elsif (ckser'event and ckser = '1') then
    hk_din1 <= q4(7) and abil0 and abil1;
  end if;
end process;

process (ckser, rst)
begin
  if (rst = '1') then
    hk_din <= '0';
  elsif (ckser'event and ckser = '1') then
    hk_din <= hk_din1;
  end if;
end process;

process (ckser, rst)
variable s : std_logic_vector (11 downto 0);
begin
  if (rst = '1') then
    s := (others => '0');
  elsif (ckser'event and ckser = '0') then
    s := st & s(11 downto 1);
  end if;
  st1 <= s(0);
end process;

process (ckser, rst)
begin
  if (rst = '1') then
    irq0 <= '1';
  elsif (ckser'event and ckser = '1') then
    irq0 <= abil0 and abil1;
  end if;
end process;

process (st1, rst)
begin
  if (rst = '1') then
    addrw(0) <= '0';
    addrw(1) <= '0';
    addrw(2) <= '0';
  elsif (st1'event and st1 = '1') then
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 67/131
Filename: *ASTEDetailedDesign6*

```
        addrw(0) <= count(10);
        addrw(1) <= count(11);
        addrw(2) <= count(12);
    end if;
end process;

ckser <= count(4);
sclk1 <= not ckser;
sclk <= sclk1;
cs <= '0';

abil0 <= not count(13) and not count(14) and not count(15);
abil1 <= not count(16) and not count(17) and not count(18) and not count(19);

sstrb1 <= st1;
sstrb2 <= st1;

st <= abil0 and abil1 and not count(5) and not count(6) and not count(7) and
not count(8) and count(9);

strb <= st1;

stx <= st;
end arc_hk;
```

5.1.14. hk_adc.vhd

```
-- hk_adc.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity hk_adc is
    port ( dout          : in  std_logic;
          sstrb         : in  std_logic;
           sclk          : in  std_logic;
           data          : out std_logic_vector (11 downto 0);
           we            : out std_logic );
end hk_adc;

architecture arc_hk_adc of hk_adc is

    constant maxcount      : unsigned (3 downto 0) := "1100";

    signal sel              : std_logic := '0';
    signal q                : std_logic_vector (11 downto 0);
    signal q1               : unsigned (3 downto 0);

begin

    process (sclk, sel)
    begin
        if (sclk'event and sclk = '0') then
```

```
        if (sel = '0') then
            q <= q(10 downto 0) & dout;
        end if;
    end if;
end process;

process(sclk, sstrb, sel)
begin
    if (sclk'event and sclk = '1') then
        if sstrb = '1' then
            q1 <= (others => '0');
        elsif sel = '0' then
            q1 <= q1 + 1;
        end if;
    end if;
end process;

process (q1)
begin
    if q1 = maxcount then
        sel <= '1';
    else
        sel <= '0';
    end if;
end process;

we <= sel;
data <= q;
```

end arc_hk_adc;

5.1.15. reg_sram.vhd

-- sram.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity reg_sram is
    port ( data1      : in      std_logic_vector (11 downto 0);
          data2      : in      std_logic_vector (11 downto 0);
          q           : out     std_logic_vector (15 downto 0);
          we1         : in      std_logic;
          we2         : in      std_logic;
          address     : in      std_logic_vector (3 downto 0);
          addrw       : in      std_logic_vector (2 downto 0);
          dms3        : in      std_logic );
end reg_sram;

architecture arc_reg_sram of reg_sram is

    type mem1 is array (0 to 7) of std_logic_vector (11 downto 0);
    type mem2 is array (0 to 7) of std_logic_vector (11 downto 0);
```

```

signal      ram_tmp1   : mem1;
signal      ram_tmp2   : mem2;
signal      addr       : std_logic_vector (2 downto 0);

begin

  process (we1)
  begin
    if we1'event and we1 = '1' then
      ram_tmp1 (conv_integer (addrw(2 downto 0))) <= data1;
    end if;
  end process;

  process (we2)
  begin
    if we2'event and we2 = '1' then
      ram_tmp2 (conv_integer (addrw(2 downto 0))) <= data2;
    end if;
  end process;

  process (dms3, address, addr)
  begin
    if dms3 = '0' then
      case address(3) is
        when '0' => q <= "0000" & ram_tmp1 (conv_integer(addr));
        when '1' => q <= "0000" & ram_tmp2 (conv_integer(addr));
        when others => q <= (others => '0');
      end case;
    end if;
  end process;

  addr <= address(2) & address(1) & address(0);

end arc_reg_sram;

```

5.1.16. bus_cont.vhd

```

-- bus_cont.vhd

library ieee;
use ieee.std_logic_1164.all;

entity bus_cont is
  port ( rst           : in      std_logic;
         ck            : in      std_logic;
         dms1          : in      std_logic;
         dms2          : in      std_logic;
         dms3          : in      std_logic;
         dmr           : in      std_logic;
         dmwr          : in      std_logic;
         flag2         : in      std_logic;
         im_we         : in      std_logic;
         data_dsp_in   : in      std_logic_vector (7 downto 0);
         addr_dsp      : in      std_logic_vector (5 downto 4);
         end_im        : in      std_logic;

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 70/131
Filename: *ASTEDetailedDesign6*

```
inttx1      : in      std_logic;
intrx1      : in      std_logic;
inttx2      : in      std_logic;
intrx2      : in      std_logic;
sel_data_dsp : out     std_logic;
sel_data_cam : out     std_logic;
sel_data_camx : out    std_logic;
sel_cam     : out     std_logic;
bitl6       : out     std_logic;
irq1        : out     std_logic;
irq3        : out     std_logic;
dir         : out     std_logic;
ram1_we     : out     std_logic;
ram1_oe     : out     std_logic;
ram1_cs     : out     std_logic;
ram2_we     : out     std_logic;
ram2_oe     : out     std_logic;
ram2_cs     : out     std_logic;
write_ser1  : out     std_logic;
read_ser1   : out     std_logic;
write_ser2  : out     std_logic;
read_ser2   : out     std_logic;
cam_we      : out     std_logic );
```

```
end bus_cont;
```

```
architecture arc_bus_cont of bus_cont is
```

```
signal sell      : std_logic;
signal sel2      : std_logic;
signal ram       : std_logic;
signal q         : std_logic;
signal ram1      : std_logic;
signal ram2      : std_logic;
signal sel_data_cams : std_logic;
signal sel_data_cam1 : std_logic;
signal sel_data_cam2 : std_logic;
signal sel_data_cam3 : std_logic;
signal sel_cams  : std_logic;
signal cam_wes  : std_logic;
signal flag2in  : std_logic;
```

```
begin
```

```
process(end_im, rst, sell, dmwr)
begin
  if ((end_im = '1') or (rst = '1')) then
    q <= '0';
  elsif dmwr'event and dmwr = '1' then
    case sell is
      when '1' => q <= '1';
      when others =>
    end case;
  end if;
end process;
```

```
process(rst, sel2, dmwr)
begin
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 71/131
Filename: ASTEDetailedDesign6

```
if rst = '1' then
    sel_cams <= '0';
elseif (dmwr'event and dmwr = '1') then
    case sel2 is
        when '1' => sel_cams <= data_dsp_in(7);
        when others =>
            end case;
    end if;
end process;

process(rst, sel1, dmwr)
begin
    if rst = '1' then
        cam_wes <= '1';
    else
        case sel1 is
            when '0' => cam_wes <= '1';
            when '1' => cam_wes <= dmwr;
            when others =>
                end case;
        end if;
    end process;

process (flag2in, ram, im_we)
begin
    case flag2in is
        when '0' =>
            ram1 <= ram;
            ram2 <= im_we;
        when '1' =>
            ram1 <= im_we;
            ram2 <= ram;
        when others =>
            ram1 <= ram1;
            ram2 <= ram2;
        end case;
    end process;

sel1 <= not dms3 and addr_dsp(5) and addr_dsp(4);
sel2 <= not dms3 and addr_dsp(5) and not addr_dsp(4);
ram <= not (not dms2 and not dmwr);

ram1_we <= ram1 or sel_data_cams;
ram2_we <= ram2 or sel_data_cams;

ram1_oe <= not (not dms2 and not dmrdr and not flag2in) or sel_data_cams;
ram2_oe <= not (not dms2 and not dmrdr and flag2in) or sel_data_cams;

ram1_cs <= not ((not dms2 and not flag2in) or (flag2in and not im_we)) or
sel_data_cams;
ram2_cs <= not ((not dms2 and flag2in) or (not flag2in and not im_we)) or
sel_data_cams;

sel_data_dsp <= not dmrdr and (not dms2 or not dms3);
sel_data_cams <= addr_dsp(5) and addr_dsp(4) and not dms3;

sel_cam <= sel_cams;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 72/131
Filename: *ASTEDetailedDesign6*

```
cam_we <= cam_wes;

write_ser1 <= not (not dms3 and not dmwr and (not addr_dsp(5) and not
addr_dsp(4)) );
write_ser2 <= not (not dms3 and not dmwr and (not addr_dsp(5) and
addr_dsp(4)) );
read_ser1 <= not (not dms3 and not dmrdr and (not addr_dsp(5) and not
addr_dsp(4)) );
read_ser2 <= not (not dms3 and not dmrdr and (not addr_dsp(5) and addr_dsp(4))
);

irq1 <= inttx1 or inttx2;
irq3 <= not (not intrx1 or not intrx2);

bit16 <= not ((not dms1 or not dms2 or not dms3) and not dmrdr);

sel_data_cam1 <= sel_data_cams or not q;
dir <= not sel_data_cam1;

process(rst, ck)
begin
  if rst = '1' then
    sel_data_cam2 <= '0';
  elsif ck'event and ck = '1' then
    sel_data_cam2 <= sel_data_cam1;
  end if;
end process;

process(rst, ck)
begin
  if rst = '1' then
    sel_data_cam3 <= '0';
  elsif ck'event and ck = '1' then
    sel_data_cam3 <= sel_data_cam2;
  end if;
end process;

sel_data_cam <= sel_data_cam1 and sel_data_cam3;
sel_data_camx <= sel_data_cams;

process(rst, ck)
begin
  if rst = '1' then
    flag2in <= '0';
  elsif ck'event and ck = '1' then
    flag2in <= flag2;
  end if;
end process;

end arc_bus_cont;
```

5.1.17. timeout.vhd

-- timeout.vhd



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 73/131
Filename: ASTEDetailedDesign6

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity timeout is
    port ( rst           : in   std_logic;
          ck            : in   std_logic;
          flag3         : in   std_logic;
          rst_cpu       : out  std_logic );
end timeout;

architecture arc_timeout of timeout is

    signal reset        :   std_logic;
    signal ck_1ms       :   std_logic;
    signal flag3q       :   std_logic;
    constant count      :   integer range 0 to 65535 := 20000;
    constant count1     :   integer range 0 to 16383 := 16383;

begin

    process(rst, ck)
        variable cnt : integer range 0 to count-1 := count-1;
    begin
        if rst = '1' then
            ck_1ms <= '0';
            cnt := count-1;
        elsif (ck'event and ck = '1') then
            if cnt = 0 then
                ck_1ms <= '1';
                cnt := count - 1;
            else
                ck_1ms <= '0';
                cnt := cnt - 1;
            end if;
        end if;
    end process;

    process(rst, reset, flag3)
    begin
        if (rst = '1' or reset = '1') then
            flag3q <= '0';
        elsif (flag3'event and flag3 = '1') then
            flag3q <= '1';
        end if;
    end process;

    process(rst, ck_1ms)
    begin
        if rst = '1' then
            reset <= '0';
        elsif (ck_1ms'event and ck_1ms = '1') then
            reset <= flag3q;
        end if;
    end process;
```

```
process(rst, reset, ck_lms)
variable cnt : integer range 0 to count1-1 := count1-1;
begin
  if (rst = '1' or reset = '1') then
    rst_cpu <= '0';
    cnt := count1-1;
  elsif (ck_lms'event and ck_lms = '1') then
    if cnt = 0 then
      rst_cpu <= '1';
      cnt := count1-1;
    else
      rst_cpu <= '0';
      cnt := cnt - 1;
    end if;
  end if;
end process;

end arc_timeout;
```

5.2. Receiver

5.2.1. top_level.vhd

```
-- top_level.vhd

library ieee;
use ieee.std_logic_1164.all;

entity top_level is
    port ( ck40_p           : in           std_logic;
          rst_p           : in           std_logic;
          sync_p          : in           std_logic;

          dricout0_p      : inout        std_logic;
          dricout1_p      : inout        std_logic;
          dricout2_p      : inout        std_logic;
          dricout3_p      : inout        std_logic;
          dricout4_p      : inout        std_logic;
          dricout5_p      : inout        std_logic;
          dricout6_p      : inout        std_logic;
          dricout7_p      : inout        std_logic;
          dricout8_p      : inout        std_logic;
          dricout9_p      : inout        std_logic;
          dricout10_p     : inout        std_logic;
          dricout11_p     : inout        std_logic;
          dricout12_p     : inout        std_logic;
          dricout13_p     : inout        std_logic;
          dricout14_p     : inout        std_logic;
          dricout15_p     : inout        std_logic;

          imrdy_p         : out           std_logic;
          imwe_p          : out           std_logic;
          rowrst_p        : out           std_logic;
          dirin_p         : in           std_logic;
          selcam_p        : in           std_logic;

          irq2_p          : out           std_logic;

          camwe_p         : in           std_logic;

          din0_p          : in           std_logic;
          sin0_p          : in           std_logic;
          din1_p          : in           std_logic;
          sin1_p          : in           std_logic;

          ckricin0_p      : in           std_logic;
          ckricin1_p      : in           std_logic;

          ckric0_p        : out           std_logic;
          ckric1_p        : out           std_logic;

          dout0_p         : out           std_logic;
          sout0_p         : out           std_logic;
          dout1_p         : out           std_logic;
          sout1_p         : out           std_logic;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 76/131
Filename: ASTEDetailedDesign6

```
jp0_p      : out      std_logic;  
jp1_p      : out      std_logic;  
jp2_p      : out      std_logic;  
jp3_p      : out      std_logic;  
jp4_p      : out      std_logic;  
jp5_p      : out      std_logic;  
jp6_p      : out      std_logic;  
jp7_p      : out      std_logic;  
  
jp72_p     : out      std_logic;  
jp73_p     : out      std_logic;  
jp74_p     : out      std_logic;  
jp75_p     : out      std_logic;  
jp76_p     : out      std_logic;  
jp77_p     : out      std_logic;  
jp78_p     : out      std_logic;  
jp79_p     : out      std_logic;  
jp710_p    : out      std_logic;  
jp711_p    : out      std_logic;  
jp712_p    : out      std_logic;  
  
ck20_p     : out      std_logic );
```

```
attribute alspin : string;
```

```
attribute alspin of ck40_p      : signal is "7";  
attribute alspin of rst_p      : signal is "207";  
attribute alspin of sync_p     : signal is "3";  
attribute alspin of dricout0_p : signal is "44";  
attribute alspin of dricout1_p : signal is "45";  
attribute alspin of dricout2_p : signal is "46";  
attribute alspin of dricout3_p : signal is "47";  
attribute alspin of dricout4_p : signal is "48";  
attribute alspin of dricout5_p : signal is "49";  
attribute alspin of dricout6_p : signal is "50";  
attribute alspin of dricout7_p : signal is "51";  
attribute alspin of dricout8_p : signal is "34";  
attribute alspin of dricout9_p : signal is "35";  
attribute alspin of dricout10_p : signal is "36";  
attribute alspin of dricout11_p : signal is "37";  
attribute alspin of dricout12_p : signal is "38";  
attribute alspin of dricout13_p : signal is "39";  
attribute alspin of dricout14_p : signal is "42";  
attribute alspin of dricout15_p : signal is "43";  
  
attribute alspin of imrdy_p    : signal is "20";  
attribute alspin of imwe_p     : signal is "24";  
attribute alspin of rowrst_p   : signal is "22";  
attribute alspin of dirin_p    : signal is "31";  
attribute alspin of selcam_p   : signal is "32";  
  
attribute alspin of irq2_p     : signal is "55";  
  
attribute alspin of camwe_p    : signal is "57";  
  
attribute alspin of din0_p     : signal is "121";  
attribute alspin of sin0_p     : signal is "122";
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 77/131
Filename: ASTEDetailedDesign6

```
attribute alspin of din1_p      : signal is "124";
attribute alspin of sin1_p      : signal is "125";

attribute alspin of ckricin0_p  : signal is "96";
attribute alspin of ckricin1_p  : signal is "100";
attribute alspin of ckric0_p    : signal is "95";
attribute alspin of ckricl_p    : signal is "99";

attribute alspin of dout0_p     : signal is "109";
attribute alspin of sout0_p     : signal is "110";
attribute alspin of dout1_p     : signal is "107";
attribute alspin of sout1_p     : signal is "106";

attribute alspin of jp0_p       : signal is "160";
attribute alspin of jp1_p       : signal is "161";
attribute alspin of jp2_p       : signal is "162";
attribute alspin of jp3_p       : signal is "163";
attribute alspin of jp4_p       : signal is "165";
attribute alspin of jp5_p       : signal is "166";
attribute alspin of jp6_p       : signal is "167";
attribute alspin of jp7_p       : signal is "168";

attribute alspin of jp72_p      : signal is "195";
attribute alspin of jp73_p      : signal is "196";
attribute alspin of jp74_p      : signal is "197";
attribute alspin of jp75_p      : signal is "198";
attribute alspin of jp76_p      : signal is "199";
attribute alspin of jp77_p      : signal is "200";
attribute alspin of jp78_p      : signal is "202";
attribute alspin of jp79_p      : signal is "203";
attribute alspin of jp710_p     : signal is "204";
attribute alspin of jp711_p     : signal is "205";
attribute alspin of jp712_p     : signal is "206";

attribute alspin of ck20_p      : signal is "9";
```

end top_level;

architecture arc_top_level of top_level is

```
component div16
  port ( rst      : in      std_logic;
        ck40     : in      std_logic;
        ck2p5    : out     std_logic;
        ck20     : out     std_logic );
end component;
```

```
component div2500
  port ( rst      : in      std_logic;
        ck2p5    : in      std_logic;
        ck1k     : out     std_logic );
end component;
```

```
component rx_complete
  port ( rst      : in      std_logic;
        ck        : in      std_logic;
        din       : in      std_logic;
```

```

sin          : in      std_logic;
ckricin      : in      std_logic;
dricout      : out     std_logic_vector (15 downto 0);
ckdout       : out     std_logic;
ctrlout      : out     std_logic_vector (5 downto 0);
ckctrlout    : out     std_logic;
ckrow        : out     std_logic;
imrdy        : out     std_logic;
timeout      : out     std_logic;
ckric        : out     std_logic;
state        : out     std_logic_vector (6 downto 0);
csyncdata    : out     std_logic );
end component;

component rx_mux
  port ( dricout0      : in      std_logic_vector (15 downto 0);
         ckdout0       : in      std_logic;
         ctrlout0      : in      std_logic_vector (5 downto 0);
         ckctrlout0    : in      std_logic;
         ckrow0        : in      std_logic;
         imrdy0        : in      std_logic;
         timeout0      : in      std_logic;
         state0        : in      std_logic_vector (6 downto 0);
         dricout1      : in      std_logic_vector (15 downto 0);
         ckdout1       : in      std_logic;
         ctrlout1      : in      std_logic_vector (5 downto 0);
         ckctrlout1    : in      std_logic;
         ckrow1        : in      std_logic;
         imrdy1        : in      std_logic;
         timeout1      : in      std_logic;
         state1        : in      std_logic_vector (6 downto 0);
         selcam        : in      std_logic;
         dricout       : out     std_logic_vector (15 downto 0);
         ckdout        : out     std_logic;
         ctrlout       : out     std_logic_vector (5 downto 0);
         ckctrlout     : out     std_logic;
         ckrow         : out     std_logic;
         imrdy         : out     std_logic;
         timeout       : out     std_logic;
         state         : out     std_logic_vector (6 downto 0) );
end component;

component tx_count
  port ( rst          : in      std_logic;
         camwe       : in      std_logic;
         ck           : in      std_logic;
         syncck      : out     std_logic;
         cdn         : out     std_logic;
         ctrl        : out     std_logic_vector (13 downto 0) );
end component;

component time_tag_counter
  port ( rst          : in      std_logic;
         sync         : in      std_logic;
         ck           : in      std_logic;
         ckdout      : in      std_logic;
         ctrlout     : in      std_logic_vector (5 downto 0);

```

```

        ckctrlout      : in      std_logic;
        dataout        : out      std_logic_vector (15 downto 0);
        dricout        : in      std_logic_vector (15 downto 0);
        dirin          : in      std_logic;
        seldata        : out      std_logic;
        ck40           : in      std_logic;
        selcam         : in      std_logic;
        elonx          : out      std_logic;
        hkonx          : out      std_logic;
        efoffx         : out      std_logic );
end component;

component tx_complete
port ( rst           : in      std_logic;
      setup          : in      std_logic_vector (2 downto 0);
      camwe          : in      std_logic;
      syncck         : in      std_logic;
      cdn            : in      std_logic;
      ctrl           : in      std_logic_vector (13 downto 0);
      ck             : in      std_logic;
      ck40           : in      std_logic;
      dout           : out      std_logic;
      sout           : out      std_logic );
end component;

component inbuf
port ( pad           : in      std_logic;
      y              : out      std_logic );
end component;

component outbuf
port ( d             : in      std_logic;
      pad            : out      std_logic );
end component;

component bibuf
port ( pad           : inout   std_logic;
      d              : in      std_logic;
      e              : in      std_logic;
      y              : out      std_logic );
end component;

component tribuff
port ( d             : in      std_logic;
      e              : in      std_logic;
      pad            : out      std_logic );
end component;

signal ck40          : std_logic;
signal rst           : std_logic;
signal ck2p5         : std_logic;
signal ck20          : std_logic;
signal ck1k          : std_logic;

signal sync          : std_logic;

signal din0          : std_logic;

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 80/131
Filename: *ASTEDetailedDesign6*

```
signal sin0           : std_logic;
signal din1           : std_logic;
signal sin1           : std_logic;
signal ckricin0       : std_logic;
signal ckricin1       : std_logic;
signal dout0          : std_logic;
signal sout0          : std_logic;
signal dout1          : std_logic;
signal sout1          : std_logic;

signal dricout0       : std_logic_vector (15 downto 0);
signal ckcout0        : std_logic;
signal ctrlout0       : std_logic_vector (5 downto 0);
signal ckctrlout0     : std_logic;
signal ckrow0         : std_logic;
signal imrdy0         : std_logic;
signal timeout0       : std_logic;
signal ckric0         : std_logic;
signal state0         : std_logic_vector (6 downto 0);

signal dricout1       : std_logic_vector (15 downto 0);
signal ckcout1        : std_logic;
signal ctrlout1       : std_logic_vector (5 downto 0);
signal ckctrlout1     : std_logic;
signal ckrow1         : std_logic;
signal imrdy1         : std_logic;
signal timeout1       : std_logic;
signal ckric1         : std_logic;
signal state1         : std_logic_vector (6 downto 0);

signal selcam         : std_logic;

signal dricout        : std_logic_vector (15 downto 0);
signal ckcout         : std_logic;
signal ctrlout        : std_logic_vector (5 downto 0);
signal ckctrlout      : std_logic;
signal ckrow          : std_logic;
signal imrdy          : std_logic;
signal timeout        : std_logic;
signal state          : std_logic_vector (6 downto 0);

signal syncck         : std_logic;
signal cdn            : std_logic;
signal ctrl           : std_logic_vector (13 downto 0);
signal setup          : std_logic_vector (6 downto 0);
signal dataout        : std_logic_vector (15 downto 0);
signal seldata        : std_logic;

signal dirin          : std_logic;
signal camwe          : std_logic;
signal irq2           : std_logic;
signal imwe           : std_logic;

signal csyncdata0     : std_logic;
signal csyncdata1     : std_logic;

signal elonx          : std_logic;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 81/131
Filename: *ASTEDetailedDesign6*

```
signal hkonx          : std_logic;
signal elhk           : std_logic;
signal efoffx        : std_logic;

begin
  inst1 : div16
    port map ( rst=>rst, ck40=>ck40, ck2p5=>ck2p5, ck20=>ck20 );

  inst2 : div2500
    port map ( rst=>rst, ck2p5=>ck2p5, ck1k=>ck1k );

  inst3a : rx_complete
    port map ( rst=>rst, ck=>ck2p5, din=>din0, sin=>sin0, ckricin=>ckricin0,
              dricout(15 downto 0)=>dricout0(15 downto 0), ckcout=>ckcout0,
              ctrlout(5 downto 0)=>ctrlout0(5 downto 0),
ckctrlout=>ckctrlout0,
              ckrow=>ckrow0, imrdy=>imrdy0, timeout=>timeout0, ckric=>ckric0,
              state(6 downto 0)=>state0(6 downto 0), csyncdata=>csyncdata0 );

  inst3b : rx_complete
    port map ( rst=>rst, ck=>ck2p5, din=>din1, sin=>sin1, ckricin=>ckricin1,
              dricout(15 downto 0)=>dricout1(15 downto 0), ckcout=>ckcout1,
              ctrlout(5 downto 0)=>ctrlout1(5 downto 0),
ckctrlout=>ckctrlout1,
              ckrow=>ckrow1, imrdy=>imrdy1, timeout=>timeout1, ckric=>ckric1,
              state(6 downto 0)=>state1(6 downto 0), csyncdata=>csyncdata1 );

  inst4 : rx_mux
    port map ( dricout0(15 downto 0)=>dricout0(15 downto 0), ckcout0=>ckcout0,
              ctrlout0(5 downto 0)=>ctrlout0(5 downto 0),
ckctrlout0=>ckctrlout0,
              ckrow0=>ckrow0, imrdy0=>imrdy0, timeout0=>timeout0,
              state0(6 downto 0)=>state0(6 downto 0),
              dricout1(15 downto 0)=>dricout1(15 downto 0), ckcout1=>ckcout1,
              ctrlout1(5 downto 0)=>ctrlout1(5 downto 0),
ckctrlout1=>ckctrlout1,
              ckrow1=>ckrow1, imrdy1=>imrdy1, timeout1=>timeout1,
              state1(6 downto 0)=>state1(6 downto 0),
              selcam=>selcam,
              dricout(15 downto 0)=>dricout(15 downto 0), ckcout=>ckcout,
              ctrlout(5 downto 0)=>ctrlout(5 downto 0), ckctrlout=>ckctrlout,
              ckrow=>ckrow, imrdy=>imrdy, timeout=>timeout,
              state(6 downto 0)=>state(6 downto 0) );

  inst5 : tx_count
    port map ( rst=>rst, camwe=>camwe, ck=>ck2p5, syncck=>syncck, cdn=>cdn,
              ctrl(13 downto 0)=>ctrl(13 downto 0) );

  inst6 : time_tag_counter
    port map ( rst=>rst, sync=>sync, ck=>ck1k, ckcout=>ckcout,
              ctrlout(5 downto 0)=>ctrlout(5 downto 0),
              ckctrlout=>ckctrlout, dataout(15 downto 0)=>dataout(15 downto
0),
              dricout(15 downto 0)=>dricout(15 downto 0),
              dirin=>dirin, seldata=>seldata, ck40=>ck40, selcam=>selcam,
              elonx=>elonx, hkonx=>hkonx, efoffx=>efoffx );
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 82/131
Filename: ASTEDetailedDesign6

```
inst7a : tx_complete
  port map ( rst=>rst, setup(2 downto 0)=>setup(2 downto 0), camwe=>camwe,
            syncck=>syncck, cdn=>cdn,
            ctrl(13 downto 0)=>ctrl(13 downto 0),
            ck=>ck2p5, ck40=>ck40, dout=>dout0, sout=>sout0 );

inst7b : tx_complete
  port map ( rst=>rst, setup(2 downto 0)=>setup(5 downto 3), camwe=>camwe,
            syncck=>syncck, cdn=>cdn,
            ctrl(13 downto 0)=>ctrl(13 downto 0),
            ck=>ck2p5, ck40=>ck40, dout=>dout1, sout=>sout1 );

inst8 : inbuf
  port map ( pad=>ck40_p, y=>ck40 );
inst9 : inbuf
  port map ( pad=>rst_p, y=>rst );
inst10 : inbuf
  port map ( pad=>sync_p, y=>sync );

inst11 : bibuf
  port map ( pad=>dricout0_p, d=>dataout(0),
            e=>seldata, y=>setup(0) );
inst12 : bibuf
  port map ( pad=>dricout1_p, d=>dataout(1),
            e=>seldata, y=>setup(1) );
inst13 : bibuf
  port map ( pad=>dricout2_p, d=>dataout(2),
            e=>seldata, y=>setup(2) );
inst14 : bibuf
  port map ( pad=>dricout3_p, d=>dataout(3),
            e=>seldata, y=>setup(3) );
inst15 : bibuf
  port map ( pad=>dricout4_p, d=>dataout(4),
            e=>seldata, y=>setup(4) );
inst16 : bibuf
  port map ( pad=>dricout5_p, d=>dataout(5),
            e=>seldata, y=>setup(5) );
inst17 : bibuf
  port map ( pad=>dricout6_p, d=>dataout(6),
            e=>seldata, y=>setup(6) );

inst18 : tribuff
  port map ( pad=>dricout7_p, d=>dataout(7), e=>seldata );
inst19 : tribuff
  port map ( pad=>dricout8_p, d=>dataout(8), e=>seldata );
inst20 : tribuff
  port map ( pad=>dricout9_p, d=>dataout(9), e=>seldata );
inst21 : tribuff
  port map ( pad=>dricout10_p, d=>dataout(10), e=>seldata );
inst22 : tribuff
  port map ( pad=>dricout11_p, d=>dataout(11), e=>seldata );
inst23 : tribuff
  port map ( pad=>dricout12_p, d=>dataout(12), e=>seldata );
inst24 : tribuff
  port map ( pad=>dricout13_p, d=>dataout(13), e=>seldata );
inst25 : tribuff
  port map ( pad=>dricout14_p, d=>dataout(14), e=>seldata );
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 83/131
Filename: *ASTEDetailedDesign6*

```
inst26 : tribuff
  port map ( pad=>dricout15_p , d=>dataout(15), e=>seldata );

inst27 : outbuf
  port map ( d=>imrdy, pad=>imrdy_p );
inst28 : outbuf
  port map ( d=>imwe, pad=>imwe_p );
inst29 : outbuf
  port map ( d=>ckrow, pad=>rowrst_p );

inst30 : inbuf
  port map ( pad=>dirin_p , y=>dirin );
inst31 : inbuf
  port map ( pad=>selcam_p , y=>selcam );

inst34 : outbuf
  port map ( d=>irq2, pad=>irq2_p );

inst36 : inbuf
  port map ( pad=>camwe_p , y=>camwe );

inst39 : inbuf
  port map ( pad=>din0_p , y=>din0 );
inst40 : inbuf
  port map ( pad=>sin0_p , y=>sin0 );
inst41 : inbuf
  port map ( pad=>din1_p , y=>din1 );
inst42 : inbuf
  port map ( pad=>sin1_p , y=>sin1 );

inst43 : outbuf
  port map ( d=>ckric0, pad=>ckric0_p );
inst44 : outbuf
  port map ( d=>ckric1, pad=>ckric1_p );

inst43a : inbuf
  port map ( pad=>ckricin0_p , y=>ckricin0 );
inst44a : inbuf
  port map ( pad=>ckricin1_p , y=>ckricin1 );

inst45 : outbuf
  port map ( d=>dout0, pad=>dout0_p );
inst46 : outbuf
  port map ( d=>sout0, pad=>sout0_p );
inst47 : outbuf
  port map ( d=>dout1, pad=>dout1_p );
inst48 : outbuf
  port map ( d=>sout1, pad=>sout1_p );

inst50 : outbuf
  port map ( d=>state(0), pad=>jp0_p );
inst51 : outbuf
  port map ( d=>state(1), pad=>jp1_p );
inst52 : outbuf
  port map ( d=>state(2), pad=>jp2_p );
inst53 : outbuf
  port map ( d=>state(3), pad=>jp3_p );
```

```
inst54 : outbuf
  port map ( d=>state(4), pad=>jp4_p );
inst55 : outbuf
  port map ( d=>state(5), pad=>jp5_p );
inst56 : outbuf
  port map ( d=>state(6), pad=>jp6_p );
inst57 : outbuf
  port map ( d=>camwe, pad=>jp7_p );

inst60 : outbuf
  port map ( d=>din1, pad=>jp72_p );
inst61 : outbuf
  port map ( d=>ckricin1, pad=>jp73_p );
inst62 : outbuf
  port map ( d=>csyncdata1, pad=>jp74_p );
inst63 : outbuf
  port map ( d=>ctrlout(2), pad=>jp75_p );
inst64 : outbuf
  port map ( d=>ctrlout(3), pad=>jp76_p );
inst65 : outbuf
  port map ( d=>ctrlout(4), pad=>jp77_p );
inst66 : outbuf
  port map ( d=>ctrlout(5), pad=>jp78_p );
inst67 : outbuf
  port map ( d=>elhk, pad=>jp79_p );
inst68 : outbuf
  port map ( d=>efoffx, pad=>jp710_p );
inst69 : outbuf
  port map ( d=>ckdout, pad=>jp711_p );
inst70 : outbuf
  port map ( d=>ckctrlout, pad=>jp712_p );

inst80: outbuf
  port map ( d=>ck20, pad=>ck20_p );

irq2 <= not imrdy and not timeout;
imwe <= ckdout or not dirin;
elhk <= elonx or hkox;
```

```
end arc_top_level;
```

5.2.2. div16.vhd

```
-- div16.vhd
```

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity div16 is
  port ( rst          : in    std_logic;
         ck40         : in    std_logic;
         ck2p5       : out   std_logic;
         ck20        : out   std_logic );
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 85/131
Filename: *ASTEDetailedDesign6*

```
end div16;

architecture arc_div16 of div16 is

    signal q      : std_logic_vector(3 downto 0);
    signal ck200  : std_logic;

begin

    process(ck40, rst)
    begin
        if rst = '0' then
            q <= (others => '0');
        elsif (ck40'event and ck40 = '1') then
            q <= q + '1';
        end if;
    end process;

    process(ck40)
    begin
        if (ck40'event and ck40 = '1') then
            ck200 <= not ck200;
        end if;
    end process;

    ck2p5 <= q(3);
    ck20 <= ck200;

end arc_div16;
```

5.2.3. div2500.vhd

```
-- div2500.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity div2500 is
    port ( rst          : in      std_logic;
          ck2p5        : in      std_logic;
          clk          : out     std_logic );
end div2500;

architecture arc_div2500 of div2500 is

    signal q          : std_logic_vector (11 downto 0);

begin

    process(ck2p5, rst)
    begin
        if rst = '0' then
            q <= (others => '0');
```

```

elsif (ck2p5'event and ck2p5 = '1') then
  if q = "100111000011" then
    q <= (others => '0');
  else q <= q + '1';
  end if;
end if;
end process;

cklk <= q(11);

end arc_div2500;

```

5.2.4. rx_complete.vhd

```

-- rx_complete.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity rx_complete is
  port ( rst          : in      std_logic;
        ck           : in      std_logic;
        din          : in      std_logic;
        sin          : in      std_logic;
        ckricin     : in      std_logic;
        dricout      : out     std_logic_vector (15 downto 0);
        ckdout       : out     std_logic;
        ctrlout      : out     std_logic_vector (5 downto 0);
        ckctrlout   : out     std_logic;
        ckrow        : out     std_logic;
        imrdy        : out     std_logic;
        timeout      : out     std_logic;
        ckric        : out     std_logic;
        state        : out     std_logic_vector (6 downto 0);
        csyncdata    : out     std_logic );
end rx_complete;

architecture arc_rx_complete of rx_complete is

  component rx_read
    port ( rst          : in      std_logic;
          din          : in      std_logic;
          ck           : in      std_logic;
          syncdata     : in      std_logic;
          dric         : out     std_logic_vector (15 downto 0) );
  end component;

  component rx_ck_rec
    port ( din         : in      std_logic;
          sin         : in      std_logic;
          ck          : in      std_logic;
          rst         : in      std_logic;
          ckric       : out     std_logic;
          syncr       : out     std_logic );
  end component;

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 87/131
Filename: ASTEDetailedDesign6

```
end component;

component rx_sync_gen
  port ( ck      : in    std_logic;
        rst      : in    std_logic;
        syncr    : in    std_logic;
        syncdata : out   std_logic;
        ckread   : out   std_logic );
end component;

component rx_data_rec
  port ( ck      : in    std_logic;
        dric     : in    std_logic_vector (15 downto 0);
        syncdata : in    std_logic;
        rst      : in    std_logic;
        parityout : out   std_logic;
        d        : out   std_logic_vector (15 downto 0);
        ckcout   : out   std_logic;
        ctrlout  : out   std_logic_vector (5 downto 0);
        ckctrlout : out   std_logic;
        ckread   : in    std_logic );
end component;

component rx_ctrl_dec
  port ( rst      : in    std_logic;
        ctrlout  : in    std_logic_vector (5 downto 0);
        ckctrlout : in    std_logic;
        ckrow    : out   std_logic );
end component;

component rx_err_ctrl_machine
  port ( rst      : in    std_logic;
        ctrlout  : in    std_logic_vector (5 downto 0);
        ckctrlout : in    std_logic;
        ck       : in    std_logic;
        image    : out   std_logic;
        state    : out   std_logic_vector (6 downto 0) );
end component;

component rx_parity_counter
  port ( rst      : in    std_logic;
        ctrlout  : in    std_logic_vector (5 downto 0);
        ckctrlout : in    std_logic;
        parityout : in    std_logic;
        p        : out   std_logic_vector (15 downto 0) );
end component;

component rx_timeout_ctrl
  port ( rst      : in    std_logic;
        ctrlout  : in    std_logic_vector (5 downto 0);
        ckctrlout : in    std_logic;
        ck       : in    std_logic;
        time     : out   std_logic );
end component;

component rx_data_mux
  port ( rst      : in    std_logic;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 88/131
Filename: ASTEDetailedDesign6

```
    d      : in      std_logic_vector (15 downto 0);  
    p      : in      std_logic_vector (15 downto 0);  
    ctrlout : in      std_logic_vector (5  downto 0);  
    ckctrlout : in    std_logic;  
    ckcout  : in      std_logic;  
    dricout : out    std_logic_vector (15 downto 0) );  
end component;
```

```
signal syncr      : std_logic;  
signal syncdata   : std_logic;  
signal dric       : std_logic_vector (15 downto 0);  
signal parityout  : std_logic;  
signal d          : std_logic_vector (15 downto 0);  
signal ckcouts    : std_logic;  
signal ctrlouts   : std_logic_vector (5  downto 0);  
signal ckctrlouts : std_logic;  
signal ckrows     : std_logic;  
signal p          : std_logic_vector (15 downto 0);  
signal ckread     : std_logic;
```

begin

```
inst1 : rx_read  
  port map ( rst=>rst, din=>din, ck=>ckricin, syncdata=>syncdata,  
            dric(15 downto 0)=>dric(15 downto 0) );  
  
inst2 : rx_ck_rec  
  port map ( din=>din, sin=>sin, ck=>ckricin, rst=>rst,  
            ckric=>ckric, syncr=>syncr );  
  
inst3 : rx_sync_gen  
  port map ( ck=>ckricin, rst=>rst, syncr=>syncr, syncdata=>syncdata,  
            ckread=>ckread );  
  
inst4 : rx_data_rec  
  port map ( ck=>ckricin,  
            dric(15 downto 0)=>dric(15 downto 0), syncdata=>syncdata,  
            rst=>rst, parityout=>parityout, d(15 downto 0)=>d(15 downto 0),  
            ckcout=>ckcouts, ctrlout(5 downto 0)=>ctrlouts(5 downto 0),  
            ckctrlout=>ckctrlouts, ckread=>ckread );  
  
inst5 : rx_ctrl_dec  
  port map ( rst=>rst, ctrlout(5 downto 0)=>ctrlouts(5 downto 0),  
            ckctrlout=>ckctrlouts, ckrow=>ckrows );  
  
inst6 : rx_err_ctrl_machine  
  port map ( rst=>rst, ctrlout(5 downto 0)=>ctrlouts(5 downto 0),  
            ckctrlout=>ckctrlouts, ck=>ck, image=>imrdy,  
            state(6 downto 0)=>state(6 downto 0) );  
  
inst7 : rx_parity_counter  
  port map ( rst=>rst, ctrlout(5 downto 0)=>ctrlouts(5 downto 0),  
            ckctrlout=>ckctrlouts, parityout=>parityout,  
            p(15 downto 0)=>p(15 downto 0) );  
  
inst8 : rx_timeout_ctrl  
  port map ( rst=>rst, ctrlout(5 downto 0)=>ctrlouts(5 downto 0),
```

```
ckctrlout=>ckctrlouts, ck=>ck, time=>timeout );

inst9 : rx_data_mux
  port map ( rst=>rst, d(15 downto 0)=>d(15 downto 0), p(15 downto 0)=>p(15
downto 0),
           ctrlout=>ctrlouts, ckctrlout=>ckctrlouts, ckdout=>ckdouts,
           dricout(15 downto 0)=>dricout(15 downto 0) );

ckdout <= ckdouts;
ctrlout <= ctrlouts;
ckctrlout <= ckctrlouts;
ckrow <= ckrows;
csyncdata <= syncdata;

end arc_rx_complete;
```

5.2.5. rx_read.vhd

```
-- rx_read.vhd

library ieee;
use ieee.std_logic_1164.all;

entity rx_read is
  port ( rst      : in    std_logic;
        din      : in    std_logic;
        ck       : in    std_logic;
        syncdata : in    std_logic;
        dric     : out   std_logic_vector (15 downto 0) );
end rx_read;

architecture arc_rx_read of rx_read is

  signal q1      : std_logic_vector (7 downto 0);
  signal q2      : std_logic_vector (7 downto 0);
  signal ck1     : std_logic;
  signal ck2     : std_logic;
  signal syncdata1 : std_logic;
  signal syncdata2 : std_logic;
  signal dric1   : std_logic_vector (15 downto 0);

begin

  process (rst, ck1, din)
  begin
    if rst = '0' then
      q1 <= (others => '0');
    elsif (ck1'event and ck1 = '1') then
      q1 <= q1(6 downto 0) & din;
    end if;
  end process;

  process (rst, ck2, din)
  begin
    if rst = '0' then
      q2 <= (others => '0');
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 90/131
Filename: *ASTEDetailedDesign6*

```
    elsif (ck2'event and ck2 = '1') then
        q2 <= q2(6 downto 0) & din;
    end if;
end process;

process (rst, syncdata1)
begin
    if rst = '0' then
        for i in 0 to 7 loop
            dric1(2*i) <= '0';
        end loop;
    elsif (syncdata1'event and syncdata1 = '1') then
        for i in 0 to 7 loop
            dric1(2*i) <= q2(7-i);
        end loop;
    end if;
end process;

process (rst, syncdata2)
begin
    if rst = '0' then
        for i in 0 to 7 loop
            dric1(2*i+1) <= '0';
        end loop;
    elsif (syncdata2'event and syncdata2 = '1') then
        for i in 0 to 7 loop
            dric1(2*i+1) <= q1(7-i);
        end loop;
    end if;
end process;

process (rst, syncdata)
begin
    if rst = '0' then
        dric <= (others => '0');
    elsif (syncdata'event and syncdata = '1') then
        dric <= dric1;
    end if;
end process;

ck1 <= not ck;
ck2 <= ck;
syncdata1 <= syncdata;
syncdata2 <= not syncdata;

end arc_rx_read;
```

5.2.6. rx_ck_rec.vhd

```
-- rx_ck_rec.vhd

library ieee;
use ieee.std_logic_1164.all;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 91/131
Filename: ASTEDetailedDesign6

```
entity rx_ck_rec is
    port ( din      : in    std_logic;
          sin      : in    std_logic;
          ck       : in    std_logic;
          rst      : in    std_logic;
          ckric    : out   std_logic;
          syncr    : out   std_logic );
end rx_ck_rec;

architecture arc_rx_ck_rec of rx_ck_rec is

    component my_and7
        port( data      : in std_logic_vector(6 downto 0);
              result    : out std_logic) ;
    end component;

    component my_and8
        port( data      : in std_logic_vector(7 downto 0);
              result    : out std_logic) ;
    end component;

    signal    ck1      : std_logic;
    signal    ck2      : std_logic;
    signal    a1       : std_logic;
    signal    a2       : std_logic;
    signal    q1       : std_logic_vector (6 downto 0);
    signal    q17      : std_logic;
    signal    q2       : std_logic_vector (7 downto 0);

begin

    inst2 : my_and7 port map ( data => q1, result => a1);
    inst3 : my_and8 port map ( data => q2, result => a2);

    process(ck1, rst)
    begin
        if rst = '0' then
            q1 <= (others => '0');
        elsif (ck1'event and ck1 = '1') then
            q1 <= q1(5 downto 0) & din;
        end if;
    end process;

    process(ck1, rst)
    begin
        if rst = '0' then
            q17 <= '1';
        elsif (ck1'event and ck1 = '1') then
            q17 <= not q1(6);
        end if;
    end process;

    process(ck2, rst)
    begin
        if rst = '0' then
            q2 <= (others => '0');
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 92/131
Filename: *ASTEDetailedDesign6*

```
    elsif (ck2'event and ck2 = '1') then
        q2 <= q2(6 downto 0) & din;
    end if;
end process;

ck1 <= ck;
ck2 <= not ck;

ckric <= din xor sin;
syncr <= a1 and a2 and q17;

end arc_rx_ck_rec;
```

5.2.7. my_and7.vhd

```
-- Version: 6.1 6.1.1.24

library ieee;
use ieee.std_logic_1164.all;
library a54SXA;

entity my_and7 is
    port( Data : in std_logic_vector(6 downto 0); Result : out
          std_logic) ;
end my_and7;

architecture DEF_ARCH of my_and7 is

    component AND4
        port(A, B, C, D : in std_logic; Y : out std_logic) ;
    end component;

    signal AND4_0_Y : std_logic ;
    begin

    AND4_0 : AND4
        port map(A => Data(3), B => Data(4), C => Data(5), D =>
            Data(6), Y => AND4_0_Y);
    AND4_Result : AND4
        port map(A => AND4_0_Y, B => Data(2), C => Data(1), D =>
            Data(0), Y => Result);
end DEF_ARCH;
```

5.2.8. my_and8.vhd

```
-- Version: 6.1 6.1.1.24

library ieee;
use ieee.std_logic_1164.all;
library a54SXA;
```

```
entity my_and8 is
  port( Data : in std_logic_vector(7 downto 0); Result : out
        std_logic) ;
end my_and8;

architecture DEF_ARCH of my_and8 is

  component AND2
    port(A, B : in std_logic; Y : out std_logic) ;
  end component;

  component AND4
    port(A, B, C, D : in std_logic; Y : out std_logic) ;
  end component;

  signal AND4_0_Y, AND2_0_Y : std_logic ;
begin

  AND2_0 : AND2
    port map(A => Data(2), B => Data(3), Y => AND2_0_Y);
  AND4_0 : AND4
    port map(A => Data(4), B => Data(5), C => Data(6), D =>
      Data(7), Y => AND4_0_Y);
  AND4_Result : AND4
    port map(A => AND4_0_Y, B => AND2_0_Y, C => Data(1), D =>
      Data(0), Y => Result);
end DEF_ARCH;
```

5.2.9. rx_sync_gen.vhd

```
-- rx_sync_gen.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity rx_sync_gen is
  port ( ck      : in    std_logic;
        rst      : in    std_logic;
        syncr    : in    std_logic;
        syncdata : out   std_logic;
        ckread   : out   std_logic );
end rx_sync_gen;

architecture arc_rx_sync_gen of rx_sync_gen is

  signal a1      : std_logic;
  signal a2      : std_logic;
  signal a3      : std_logic;
  signal q1      : std_logic;
  signal q       : std_logic_vector (2 downto 0);
  signal syncdata1 : std_logic;
```

```
begin

  process(rst, ck, syncr)
  begin
    if rst = '0' then
      q <= (others => '0');
    elsif (ck'event and ck = '1') then
      if syncr = '1' then
        q <= (others => '0');
      else q <= q + '1';
      end if;
    end if;
  end process;

  process (rst, ck)
  begin
    if rst = '0' then
      q1 <= '0';
      ckread <= '1';
    elsif (ck'event and ck = '0') then
      q1 <= a1;
      ckread <= a3 or a2;
    end if;
  end process;

  process (rst, syncdata1, syncr)
  begin
    if (rst = '0' or syncdata1 = '1') then
      a2 <= '0';
    elsif (syncr'event and syncr = '1') then
      a2 <= '1';
    end if;
  end process;

  a1 <= q(0) and q(1) and q(2);
  a3 <= q(2) and (q(1) or q(0));
  syncdata <= a1 and q1;
  syncdata1 <= a1 and q1;

end arc_rx_sync_gen;
```

5.2.10. rx_data_rec.vhd

```
-- rx_data_rec.vhd

library ieee;
use ieee.std_logic_1164.all;

entity rx_data_rec is
  port ( ck          : in    std_logic;
         dric        : in    std_logic_vector (15 downto 0);
         syncdata    : in    std_logic;
         rst         : in    std_logic;
         parityout   : out   std_logic;
         d           : out   std_logic_vector (15 downto 0);
```

```
        ckcout      : out   std_logic;
        ctrlout     : out   std_logic_vector (5 downto 0);
        ckctrlout   : out   std_logic;
        ckread      : in    std_logic );
end rx_data_rec;

architecture arc_rx_data_rec of rx_data_rec is

    component my_xor14
        port( data      : in std_logic_vector(13 downto 0);
              result    : out std_logic) ;
    end component;

    signal      a0          : std_logic;
    signal      a1          : std_logic;
    signal      a2          : std_logic;
    signal      parityerrs  : std_logic;
    signal      parityerrs1 : std_logic;
    signal      q1          : std_logic_vector (14 downto 0);
    signal      q2          : std_logic_vector (14 downto 0);
    signal      syncdata00  : std_logic;
    signal      syncdata01  : std_logic;
    signal      syncdata10  : std_logic;
    signal      syncdata11  : std_logic;
    signal      ckcouts     : std_logic;
    signal      ckcout1     : std_logic;

begin

    inst1 : my_xor14 port map ( data=>q1(14 downto 1), result=>a0);

    process (rst, syncdata00)
    begin
        if rst = '0' then
            q1(14 downto 12) <= (others => '0');
            q2(14 downto 12) <= (others => '0');
        elsif (syncdata00'event and syncdata00 = '1') then
            q1(14 downto 12) <= dric(15 downto 13);
            q2(14 downto 12) <= q1(14 downto 12);
        end if;
    end process;

    process (rst, syncdata01)
    begin
        if rst = '0' then
            q1(11 downto 8) <= (others => '0');
            q2(11 downto 8) <= (others => '0');
        elsif (syncdata01'event and syncdata01 = '1') then
            q1(11 downto 8) <= dric(12 downto 9);
            q2(11 downto 8) <= q1(11 downto 8);
        end if;
    end process;

    process (rst, syncdata10)
    begin
        if rst = '0' then
            q1(7 downto 4) <= (others => '0');
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 96/131
Filename: ASTEDetailedDesign6

```
    q2(7 downto 4) <= (others => '0');
    elsif (syncdata10'event and syncdata10 = '1') then
        q1(7 downto 4) <= dric(8 downto 5);
        q2(7 downto 4) <= q1(7 downto 4);
    end if;
end process;

process (rst, syncdata11)
begin
    if rst = '0' then
        q1(3 downto 0) <= (others => '0');
        q2(3 downto 0) <= (others => '0');
    elsif (syncdata11'event and syncdata11 = '1') then
        q1(3 downto 0) <= dric(4 downto 1);
        q2(3 downto 0) <= q1(3 downto 0);
    end if;
end process;

process (rst, syncdata00)
begin
    if rst = '0' then
        parityerrs1 <= '1';
    elsif (syncdata00'event and syncdata00 = '0') then
        parityerrs1 <= not(a0 xor dric(0) xor dric(1)) and not q1(0);
    end if;
end process;

process (rst, syncdata00)
begin
    if rst = '0' then
        parityerrs <= '1';
    elsif (syncdata00'event and syncdata00 = '1') then
        parityerrs <= parityerrs1;
    end if;
end process;

process (rst, ck)
begin
    if rst = '0' then
        ckcout1 <= '1';
    elsif (ck'event and ck = '1') then
        ckcout1 <= ckcouts;
    end if;
end process;

syncdata00 <= syncdata;
syncdata01 <= syncdata;
syncdata10 <= syncdata;
syncdata11 <= syncdata;

parityout <= parityerrs;

a1 <= q2(0);
-- a1 <= parityerrs or q2(0);
a2 <= parityerrs or not q2(0);

d(13 downto 0) <= q2(14 downto 1) when a1 = '0' else (others => '0');
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 97/131
Filename: ASTEDetailedDesign6

```
-- d(13 downto 0) <= q2(14 downto 1);  
d(14) <= '0';  
d(15) <= parityerrs;  
  
ctrlout <= q2(6 downto 1) when a2 = '0' else (others => '0');  
  
ckdouts <= q2(0) or ckread;  
  
ckctrlout <= not q2(0) or ckread;  
  
ckdout <= ckdout1 or ckread;  
  
end arc_rx_data_rec;
```

5.2.11. my_xor14.vhd

```
-- Version: 6.1 6.1.1.24  
  
library ieee;  
use ieee.std_logic_1164.all;  
library a54SXA;  
  
entity my_xor14 is  
    port( Data : in std_logic_vector(13 downto 0); Result : out  
          std_logic) ;  
end my_xor14;  
  
architecture DEF_ARCH of my_xor14 is  
  
    component CM8INV  
        port(A : in std_logic; Y : out std_logic) ;  
    end component;  
  
    component CM8F  
        port(D0, D1, D2, D3, S00, S01, S10, S11 : in std_logic;  
            Y, FY : out std_logic) ;  
    end component;  
  
    component VCC  
        port( Y : out std_logic);  
    end component;  
  
    component GND  
        port( Y : out std_logic);  
    end component;  
  
    signal Data_15_net, Data_16_net, Data_17_net, Data_14_net,  
           Data_18_net, Data_19_net, CM8INV_1_Y, CM8INV_5_Y,  
           CM8INV_2_Y, CM8INV_3_Y, CM8INV_0_Y, CM8INV_4_Y, VCC_1_net,  
           GND_1_net : std_logic ;  
  
    begin  
  
    VCC_2_net : VCC port map(Y => VCC_1_net);
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 98/131
Filename: *ASTEDetailedDesign6*

```
GND_2_net : GND port map(Y => GND_1_net);
CM8INV_2 : CM8INV
  port map(A => Data_17_net, Y => CM8INV_2_Y);
CM8F_Data_16_inst : CM8F
  port map(D0 => CM8INV_1_Y, D1 => Data(8), D2 => Data(8),
    D3 => CM8INV_1_Y, S00 => VCC_1_net, S01 => Data(6),
    S10 => Data(7), S11 => GND_1_net, Y => OPEN , FY =>
    Data_16_net);
CM8INV_5 : CM8INV
  port map(A => Data(5), Y => CM8INV_5_Y);
CM8F_Data_15_inst : CM8F
  port map(D0 => CM8INV_5_Y, D1 => Data(5), D2 => Data(5),
    D3 => CM8INV_5_Y, S00 => VCC_1_net, S01 => Data(3),
    S10 => Data(4), S11 => GND_1_net, Y => OPEN , FY =>
    Data_15_net);
CM8INV_4 : CM8INV
  port map(A => Data(11), Y => CM8INV_4_Y);
CM8F_Data_18_inst : CM8F
  port map(D0 => CM8INV_3_Y, D1 => Data_14_net, D2 =>
    Data_14_net, D3 => CM8INV_3_Y, S00 => VCC_1_net, S01 =>
    Data(12), S10 => Data(13), S11 => GND_1_net, Y => OPEN ,
    FY => Data_18_net);
CM8F_Data_17_inst : CM8F
  port map(D0 => CM8INV_4_Y, D1 => Data(11), D2 => Data(11),
    D3 => CM8INV_4_Y, S00 => VCC_1_net, S01 => Data(9),
    S10 => Data(10), S11 => GND_1_net, Y => OPEN , FY =>
    Data_17_net);
CM8F_Data_14_inst : CM8F
  port map(D0 => Data(2), D1 => CM8INV_0_Y, D2 => CM8INV_0_Y,
    D3 => Data(2), S00 => VCC_1_net, S01 => Data(0), S10 =>
    GND_1_net, S11 => Data(1), Y => OPEN , FY => Data_14_net);
CM8F_Data_19_inst : CM8F
  port map(D0 => CM8INV_2_Y, D1 => Data_17_net, D2 =>
    Data_17_net, D3 => CM8INV_2_Y, S00 => VCC_1_net, S01 =>
    Data_15_net, S10 => Data_16_net, S11 => GND_1_net, Y =>
    OPEN , FY => Data_19_net);
CM8INV_0 : CM8INV
  port map(A => Data(2), Y => CM8INV_0_Y);
CM8INV_1 : CM8INV
  port map(A => Data(8), Y => CM8INV_1_Y);
CM8INV_3 : CM8INV
  port map(A => Data_14_net, Y => CM8INV_3_Y);
CM8F_Result : CM8F
  port map(D0 => VCC_1_net, D1 => GND_1_net, D2 => GND_1_net,
    D3 => VCC_1_net, S00 => VCC_1_net, S01 => Data_18_net,
    S10 => Data_19_net, S11 => GND_1_net, Y => OPEN , FY =>
    Result);
end DEF_ARCH;
```

5.2.12. rx_ctrl_dec.vhd

```
-- rx_ctrl_dec.vhd

library ieee;
use ieee.std_logic_1164.all;
```

```
entity rx_ctrl_dec is
    port ( rst          : in    std_logic;
          ctrlout      : in    std_logic_vector (5 downto 0);
          ckctrlout    : in    std_logic;
          ckrow        : out   std_logic );
end rx_ctrl_dec;
```

```
architecture arc_rx_ctrl_dec of rx_ctrl_dec is
```

```
    signal  elon       : std_logic;
    signal  hkon       : std_logic;
    signal  eloff      : std_logic;
    signal  efon       : std_logic;
    signal  a1         : std_logic;
    signal  a2         : std_logic;
```

```
begin
```

```
    process (rst, ckctrlout, a2)
    begin
        if rst = '0' then
            ckrow <= '0';
        elsif (ckctrlout'event and ckctrlout = '0') then
            case a2 is
                when '0' => ckrow <= a1;
                when others => ckrow <= '0';
            end case;
        end if;
    end process;
```

```
    elon <= '1' when ctrlout = "010000" else '0';
    eloff <= '1' when (ctrlout = "100000") else '0';
    hkon <= '1' when ctrlout = "100100" else '0';
    efon <= '1' when (ctrlout = "000100") else '0';
```

```
    a1 <= efon or eloff;
    a2 <= elon or hkon;
```

```
end arc_rx_ctrl_dec;
```

5.2.13. rx_err_ctrl_machine.vhd

```
-- rx_err_ctrl_machine.vhd
```

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
```

```
entity rx_err_ctrl_machine is
    port ( rst          : in    std_logic;
          ctrlout      : in    std_logic_vector (5 downto 0);
          ckctrlout    : in    std_logic;
          ck           : in    std_logic;
          image        : out   std_logic;
          state        : out   std_logic_vector (6 downto 0) );
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 100/131
Filename: *ASTEDetailedDesign6*

```
end rx_err_ctrl_machine;

architecture arc_rx_err_ctrl_machine of rx_err_ctrl_machine is

    signal      efon          : std_logic;
    signal      efoff         : std_logic;
    signal      elon          : std_logic;
    signal      eloff         : std_logic;
    signal      hkon          : std_logic;
    signal      ckfsm         : std_logic;
    signal      reg           : std_logic_vector (4 downto 0);
    signal      imrdy         : std_logic;
    signal      q             : std_logic_vector (9 downto 0);
    signal      sel           : std_logic;

    type state_values is (st0, st1, st2, st3, st4, st5, st6);
    signal      pres_state, next_state: state_values;
    attribute syn_encoding : string;
    attribute syn_encoding of pres_state : signal is "safe,onehot";

begin

    process (rst, ckctrlout)
    begin
        if rst = '0' then
            reg <= (others => '0');
        elsif (ckctrlout'event and ckctrlout = '0') then
            reg(0) <= efon;
            reg(1) <= efoff;
            reg(2) <= elon;
            reg(3) <= eloff;
            reg(4) <= hkon;
        end if;
    end process;

-- fsm register

    process (rst, ckfsm)
    begin
        if rst = '0' then
            pres_state <= st0;
        elsif (ckfsm'event and ckfsm = '1') then
            pres_state <= next_state;
        end if;
    end process;

-- fsm combinational block

    process (pres_state, reg)
    begin
        case pres_state is
            when st0 =>
                case reg is
                    when "00001" =>
                        next_state <= st1;
                    when others => next_state <= st0;
                end case;
        end case;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 101/131
Filename: ASTEDetailedDesign6

```
when st1 =>
  case reg is
    when "00100" =>
      next_state <= st2;
    when others => next_state <= st0;
  end case;
when st2 =>
  case reg is
    when "01000" =>
      next_state <= st3;
    when others => next_state <= st0;
  end case;
when st3 =>
  case reg is
    when "00100" =>
      next_state <= st4;
    when "10000" =>
      next_state <= st5;
    when others => next_state <= st0;
  end case;
when st4 =>
  case reg is
    when "01000" =>
      next_state <= st3;
    when others => next_state <= st0;
  end case;
when st5 =>
  case reg is
    when "00010" =>
      next_state <= st6;
    when others => next_state <= st0;
  end case;
when st6 =>
  case reg is
    when "00001" =>
      next_state <= st1;
    when others => next_state <= st0;
  end case;
end case;
end process;

-- *****
-- output definition using pres_state only

process (pres_state)
begin
  case pres_state is
    when st0    => state <= "0000001";
    when st1    => state <= "0000010";
    when st2    => state <= "0000100";
    when st3    => state <= "0001000";
    when st4    => state <= "0010000";
    when st5    => state <= "0100000";
    when st6    => state <= "1000000";
  end case;
end process;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 102/131
Filename: ASTEDetailedDesign6

```
-- *****  
  
process (rst, ck)                                -- shift reg per imrdy  
begin  
  if rst = '0' then  
    q <= (others => '0');  
  elsif (ck'event and ck = '1') then  
    q <= q(8 downto 0) & imrdy;  
  end if;  
end process;  
  
-- *****  
  
efon <= '1' when ctrlout = "000100" else '0';  
efoff <= '1' when ctrlout = "001000" else '0';  
elon <= '1' when ctrlout = "010000" else '0';  
eloff <= '1' when ctrlout = "100000" else '0';  
hkcon <= '1' when ctrlout = "100100" else '0';  
  
process (rst, ckctrlout)  
begin  
  if rst = '0' then  
    ckfsm <= '0';  
  elsif (ckctrlout'event and ckctrlout = '1') then  
    case sel is  
      when '1' => ckfsm <= '1';  
      when others => ckfsm <= '0';  
    end case;  
  end if;  
end process;  
  
sel <= reg(0) or reg(1) or reg(2) or reg(3) or reg(4);  
imrdy <= '1' when pres_state = st6 else '0';  
  
image <= q(9);  
end arc_rx_err_ctrl_machine;
```

5.2.14. rx_parity_counter.vhd

```
-- rx_parity_counter.vhd  
  
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.std_logic_arith.all;  
use ieee.std_logic_unsigned.all;  
  
entity rx_parity_counter is  
  port ( rst          : in    std_logic;  
         ctrlout      : in    std_logic_vector (5 downto 0);  
         ckctrlout    : in    std_logic;  
         parityout    : in    std_logic;  
         p            : out   std_logic_vector (15 downto 0) );  
end rx_parity_counter;
```

architecture arc_rx_parity_counter of rx_parity_counter is

```
signal    ck0      : std_logic;
signal    ck1      : std_logic;
signal    cl1      : std_logic;
signal    cl2      : std_logic;
signal    efonq    : std_logic;
signal    elong    : std_logic;
signal    eloffq   : std_logic;
signal    a4       : std_logic;
signal    efon     : std_logic;
signal    elon     : std_logic;
signal    eloff    : std_logic;
signal    q        : std_logic_vector (8 downto 0);
```

begin

```
process(rst, ckctrlout)
begin
  if rst = '0' then
    efonq <= '0';
    elong <= '0';
    eloffq <= '0';
  elsif ckctrlout'event and ckctrlout = '0' then
    efonq <= efon;
    elong <= elon;
    eloffq <= eloff;
  end if;
end process;
```

```
process(cl2, eloffq)
begin
  if cl2 = '1' then
    a4 <= '1';
  elsif eloffq'event and eloffq = '1' then
    a4 <= '0';
  end if;
end process;
```

```
process(cl2, ck0)
begin
  if cl2 = '1' then
    ck1 <= '0';
  elsif ck0'event and ck0 = '1' then
    ck1 <= '1';
  end if;
end process;
```

```
process(cl1, ck1)
begin
  if cl1 = '1' then
    q <= (others => '0');
  elsif (ck1'event and ck1 = '1') then
    q <= q + '1';
  end if;
end process;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 104/131
Filename: ASTEDetailedDesign6

```
efon <= '1' when (ctrlout = "000100") else '0';  
elon <= '1' when (ctrlout = "010000") else '0';  
eloff <= '1' when (ctrlout = "100000") else '0';  
ck0 <= a4 and parityout;  
cl1 <= efonq or not rst;  
cl2 <= elong or not rst;  
p(8 downto 0) <= q;  
p(15 downto 9) <= (others => '0');
```

```
end arc_rx_parity_counter;
```

5.2.15. rx_timeout_ctrl.vhd

```
-- rx_timeout_ctrl.vhd
```

```
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.std_logic_arith.all;  
use ieee.std_logic_unsigned.all;  
  
entity rx_timeout_ctrl is  
    port ( rst          : in      std_logic;  
          ctrlout      : in      std_logic_vector (5 downto 0);  
          ckctrlout    : in      std_logic;  
          ck           : in      std_logic;  
          time         : out     std_logic );  
end rx_timeout_ctrl;  
  
architecture arc_rx_timeout_ctrl of rx_timeout_ctrl is  
  
    signal efon          : std_logic;  
    signal efonq         : std_logic;  
    signal q             : std_logic_vector (18 downto 0);  
  
begin  
  
    process(rst, ckctrlout)  
    begin  
        if rst = '0' then  
            efonq <= '0';  
        elsif ckctrlout'event and ckctrlout = '0' then  
            efonq <= efon;  
        end if;  
    end process;  
  
    process(rst, efonq, ck)  
    begin  
        if rst = '0' or efonq = '1' then  
            q <= (others => '0');  
        elsif (ck'event and ck = '1') then  
            q <= q + '1';  
        end if;  
    end process;  
  
    efon <= '1' when ctrlout = "000100" else '0';  
    time <= '1' when q = "111111111111111111" else '0';
```

```
end arc_rx_timeout_ctrl;
```

5.2.16. rx_data_mux.vhd

```
-- rx_data_mux.vhd
```

```
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.std_logic_arith.all;  
use ieee.std_logic_unsigned.all;
```

```
entity rx_data_mux is  
    port ( rst           : in      std_logic;  
          d             : in      std_logic_vector (15 downto 0);  
          p             : in      std_logic_vector (15 downto 0);  
          ctrlout       : in      std_logic_vector (5 downto 0);  
          ckctrlout     : in      std_logic;  
          ckcout        : in      std_logic;  
          dricout       : out     std_logic_vector (15 downto 0) );  
end rx_data_mux;
```

```
architecture arc_rx_data_mux of rx_data_mux is
```

```
    signal clr          : std_logic;  
    signal q            : std_logic_vector (4 downto 0);  
    signal hk           : std_logic;  
    signal hkong        : std_logic;  
    signal hkongq       : std_logic;  
    signal efoff        : std_logic;  
    signal efoffq       : std_logic;
```

```
begin
```

```
    process(rst, ckctrlout)  
    begin  
        if rst = '0' then  
            hkongq <= '0';  
            efoffq <= '0';  
        elsif ckctrlout'event and ckctrlout = '0' then  
            hkongq <= hkong;  
            efoffq <= efoff;  
        end if;  
    end process;
```

```
    process(rst, efoffq, hkongq)  
    begin  
        if rst = '0' or efoffq = '1' then  
            hk <= '0';  
        elsif hkongq'event and hkongq = '1' then  
            hk <= '1';  
        end if;  
    end process;
```

```
    process(clr, ckcout)  
    begin
```

```

if clr = '1' then
  q <= (others => '0');
elsif (ckdout'event and ckdout = '0') then
  case q(4) is
    when '0' => q <= q + '1';
    when others =>
      end case;
  end if;
end process;

process (q, p, d)
begin
  case q is
    when "00101" => dricout <= p;
    when others => dricout <= d;
  end case;
end process;

hkon <= '1' when (ctrlout = "100100") else '0';
efoff <= '1' when (ctrlout = "001000") else '0';

clr <= not rst or not hk;

end arc_rx_data_mux;

```

5.2.17. rx_mux.vhd

```

-- rx_mux.vhd

library ieee;
use ieee.std_logic_1164.all;

entity rx_mux is
  port ( dricout0      : in      std_logic_vector (15 downto 0);
         ckdout0      : in      std_logic;
         ctrlout0     : in      std_logic_vector (5 downto 0);
         ckctrlout0   : in      std_logic;
         ckrow0       : in      std_logic;
         imrdy0       : in      std_logic;
         timeout0     : in      std_logic;
         state0       : in      std_logic_vector (6 downto 0);
         dricout1     : in      std_logic_vector (15 downto 0);
         ckdout1     : in      std_logic;
         ctrlout1     : in      std_logic_vector (5 downto 0);
         ckctrlout1   : in      std_logic;
         ckrow1       : in      std_logic;
         imrdy1       : in      std_logic;
         timeout1     : in      std_logic;
         state1       : in      std_logic_vector (6 downto 0);
         selcam       : in      std_logic;
         dricout      : out     std_logic_vector (15 downto 0);
         ckdout       : out     std_logic;
         ctrlout      : out     std_logic_vector (5 downto 0);
         ckctrlout    : out     std_logic;
         ckrow        : out     std_logic;
         imrdy        : out     std_logic;

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 107/131
Filename: ASTEDetailedDesign6

```
        timeout      : out      std_logic;
        state         : out      std_logic_vector (6 downto 0) );
end rx_mux;

architecture arc_rx_mux of rx_mux is

    signal  dricouts   : std_logic_vector (15 downto 0);
    signal  ckdouts    : std_logic;
    signal  ckctrlouts : std_logic;
    signal  ctrlouts   : std_logic_vector (5 downto 0);
    signal  ckrows     : std_logic;
    signal  imrdys     : std_logic;
    signal  timeouts   : std_logic;
    signal  states     : std_logic_vector (6 downto 0);

begin

    process (selcam, dricout0, dricout1)
    begin
        case selcam is
            when '0' => dricouts <= dricout0;
            when '1' => dricouts <= dricout1;
            when others => dricouts <= dricouts;
        end case;
    end process;

    process (selcam, ckdout0, ckdout1)
    begin
        case selcam is
            when '0' => ckdouts <= ckdout0;
            when '1' => ckdouts <= ckdout1;
            when others => ckdouts <= ckdouts;
        end case;
    end process;

    process (selcam, ckctrlout0, ckctrlout1)
    begin
        case selcam is
            when '0' => ckctrlouts <= ckctrlout0;
            when '1' => ckctrlouts <= ckctrlout1;
            when others => ckctrlouts <= ckctrlouts;
        end case;
    end process;

    process (selcam, ctrlout0, ctrlout1)
    begin
        case selcam is
            when '0' => ctrlouts <= ctrlout0;
            when '1' => ctrlouts <= ctrlout1;
            when others => ctrlouts <= ctrlouts;
        end case;
    end process;

    process (selcam, ckrow0, ckrow1)
    begin
        case selcam is
            when '0' => ckrows <= ckrow0;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 108/131
Filename: *ASTEDetailedDesign6*

```
    when '1' => ckrows <= ckrow1;
    when others => ckrows <= ckrows;
end case;
end process;

process (selcam, imrdy0, imrdy1)
begin
    case selcam is
        when '0' => imrdys <= imrdy0;
        when '1' => imrdys <= imrdy1;
        when others => imrdys <= imrdys;
    end case;
end process;

process (selcam, timeout0, timeout1)
begin
    case selcam is
        when '0' => timeouts <= timeout0;
        when '1' => timeouts <= timeout1;
        when others => timeouts <= timeouts;
    end case;
end process;

process (selcam, state0, state1)
begin
    case selcam is
        when '0' => states <= state0;
        when '1' => states <= state1;
        when others => states <= states;
    end case;
end process;

dricout <= dricouts;
ckdout <= ckdouts;
ckctrlout <= ckctrlouts;
ctrlout <= ctrlouts;
ckrow <= ckrows;
imrdy <= imrdys;
timeout <= timeouts;
state <= states;
```

```
end arc_rx_mux;
```

5.2.18. tx_count.vhd

```
-- tx_count.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity tx_count is
    port ( rst          : in    std_logic;
          camwe        : in    std_logic;
          ck            : in    std_logic;
```

```

        syncck      : out      std_logic;
        cdn         : out      std_logic;
        ctrl        : out      std_logic_vector (13 downto 0) );
end tx_count;

architecture arc_tx_count of tx_count is

    signal    a          : std_logic;
    signal    ck0        : std_logic;
    signal    clr        : std_logic;
    signal    dec1       : std_logic;
    signal    dec2       : std_logic;
    signal    dec3       : std_logic;
    signal    dec        : std_logic;
    signal    q          : std_logic_vector (3 downto 0);

begin

    process(rst, ck)
    begin
        if rst = '0' then
            a <= '0';
        elsif (ck'event and ck = '0') then
            a <= camwe;
        end if;
    end process;

    process(clr, ck0)
    begin
        if clr = '1' then
            q <= (others => '0');
        elsif (ck0'event and ck0 = '1') then
            q <= q + '1';
        end if;
    end process;

    ck0 <= ck and not (q(3) and q(2) and q(1) and q(0));
    clr <= not rst or not a;
    dec1 <= '1' when q = "0001" else '0';
    dec2 <= '1' when q = "0010" else '0';
    dec3 <= '1' when q = "0011" else '0';
    dec <= dec1 or dec2;
    ctrl(13 downto 2) <= (others => dec2);
    ctrl(1 downto 0) <= (others => dec);
    syncck <= dec1;
    cdn <= not dec3;

end arc_tx_count;

```

5.2.19. time_tag_counter.vhd

```

-- time_tag_counter.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

```

```
use ieee.std_logic_unsigned.all;

entity time_tag_counter is
    port ( rst           : in      std_logic;
          sync          : in      std_logic;
          ck             : in      std_logic;
          ckcout        : in      std_logic;
          ctrlout       : in      std_logic_vector (5 downto 0);
          ckctrlout     : in      std_logic;
          dataout       : out     std_logic_vector (15 downto 0);
          dricout       : in      std_logic_vector (15 downto 0);
          dirin         : in      std_logic;
          seldata       : out     std_logic;
          ck40          : in      std_logic;
          selcam        : in      std_logic;
          elonx         : out     std_logic;
          hkonz         : out     std_logic;
          eoffx         : out     std_logic );
end time_tag_counter;

architecture arc_time_tag_counter of time_tag_counter is

    signal    clr1           : std_logic;
    signal    q1             : std_logic_vector (31 downto 0);
    signal    clr2           : std_logic;
    signal    ck2            : std_logic;
    signal    q2             : std_logic_vector (4 downto 0);
    signal    a              : std_logic;
    signal    q3             : std_logic_vector (3 downto 0);
    signal    signature      : std_logic_vector (15 downto 0);
    signal    selcamn       : std_logic;

    signal    elon           : std_logic;
    signal    elong         : std_logic;
    signal    eloff         : std_logic;
    signal    eloffq        : std_logic;
    signal    efon          : std_logic;
    signal    efonq         : std_logic;
    signal    eoff          : std_logic;
    signal    eoffq         : std_logic;
    signal    hkon          : std_logic;
    signal    hkonz         : std_logic;
    signal    counten       : std_logic;
    signal    counten1      : std_logic;
    signal    counten2      : std_logic;

    signal    sumin         : std_logic_vector (16 downto 0);
    signal    sumout        : std_logic_vector (16 downto 0);
    signal    sumout1       : std_logic_vector (16 downto 0);
    signal    sum           : std_logic_vector (16 downto 0);

    signal    min           : std_logic_vector (11 downto 0);
    signal    min1          : std_logic_vector (11 downto 0);
    signal    max           : std_logic_vector (11 downto 0);
    signal    max1          : std_logic_vector (11 downto 0);

begin
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 111/131
Filename: *ASTEDetailedDesign6*

```
process(clr1, ck)
begin
  if clr1 = '1' then
    q1 <= (others => '0');
  elsif (ck'event and ck = '1') then
    q1 <= q1 + '1';
  end if;
end process;

process(clr2, ck2)
begin
  if clr2 = '1' then
    q2 <= (others => '0');
  elsif (ck2'event and ck2 = '1') then
    case q2(4) is
      when '0' => q2 <= q2 + '1';
      when others =>
    end case;
  end if;
end process;

process (rst, ck40)
begin
  if rst = '0' then
    q3 <= (others => '0');
  elsif (ck40'event and ck40 = '1') then
    q3 <= q3(2 downto 0) & dirin;
  end if;
end process;

process (rst, ckctrlout)
begin
  if rst = '0' then
    elong <= '0';
    eloffq <= '0';
    efong <= '0';
    efoffq <= '0';
    hkong <= '0';
  elsif (ckctrlout'event and ckctrlout = '0') then
    elong <= elon;
    eloffq <= eloff;
    efong <= efon;
    efoffq <= efoff;
    hkong <= hkong;
  end if;
end process;

process (rst, hkong, efong)
begin
  if rst = '0' or hkong = '1' then
    counten1 <= '0';
  elsif (efong'event and efong = '1') then
    counten1 <= '1';
  end if;
end process;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 112/131
Filename: ASTEDetailedDesign6

```
process (rst, eloffq, elong)
begin
  if rst = '0' or eloffq = '1' then
    counten2 <= '0';
  elsif (elong'event and elong = '1') then
    counten2 <= '1';
  end if;
end process;

process (rst, counten, ckdout, dricout)
begin
  if rst = '0' then
    sumin <= (others => '0');
  elsif (ckdout'event and ckdout = '0') then
    case counten is
      when '1' => sumin <= "0000000000000000" & dricout(11) & dricout(10);
      when others => sumin <= (others => '0');
    end case;
  end if;
end process;

process (rst, efonq, ckdout)
begin
  if (rst = '0' or efonq = '1') then
    sumout <= (others => '0');
  elsif (ckdout'event and ckdout = '0') then
    sumout <= sum;
  end if;
end process;

process (rst, efonq, counten, ckdout, dricout)
begin
  if (rst = '0' or efonq = '1') then
    min <= (others => '1');
  elsif (ckdout'event and ckdout = '0') then
    if (dricout(11 downto 0) < min) and counten = '1' then
      min <= dricout(11 downto 0);
    else
      end if;
  end if;
end process;

process (rst, efonq, counten, ckdout, dricout)
begin
  if (rst = '0' or efonq = '1') then
    max <= (others => '0');
  elsif (ckdout'event and ckdout = '0') then
    if (dricout(11 downto 0) > max) and counten = '1' then
      max <= dricout(11 downto 0);
    else
      end if;
  end if;
end process;

process (rst, ckdout)
begin
  if rst = '0' then
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 113/131
Filename: *ASTEDetailedDesign6*

```
sumout1 <= (others => '0');
min1 <= (others => '0');
max1 <= (others => '0');
elsif (ckdout'event and ckdout = '0') then
    sumout1 <= sumout;
    min1 <= min;
    max1 <= max;
end if;
end process;

process (q2, q1, signature, sumout1, min1, max1, dricout)
begin
    case q2 is
        when "00110" => dataout <= q1(15 downto 0);
        when "00111" => dataout <= q1(31 downto 16);
        when "01000" => dataout <= signature;
        when "01001" => dataout <= sumout1 (15 downto 0);
        when "01010" => dataout <= "0000" & min1;
        when "01011" => dataout <= "0000" & max1;
        when "01100" => dataout <= "1010101010101010";
        when "01101" => dataout <= "0101010101010101";
        when others => dataout <= dricout;
    end case;
end process;

clr1 <= not rst or sync;
clr2 <= not rst or not hkong;
a <= '0' when q2 = "1110" else '1';
ck2 <= not ckdout and a;
seldata <= q3(3) and q3(0);
selcamn <= not selcam;
signature <= "0000000000000000" & selcam & selcamn;

elon <= '1' when (ctrlout = "010000") else '0';
eloff <= '1' when (ctrlout = "100000") else '0';
efon <= '1' when (ctrlout = "000100") else '0';
efoff <= '1' when (ctrlout = "001000") else '0';
hkonn <= '1' when (ctrlout = "100100") else '0';

counten <= counten1 and counten2;

sum <= sumin + sumout when sumout(16) = '0' else (others => '1');

elonx <= elong;
hkonnx <= hkonn;
efoffx <= efoffq;

end arc_time_tag_counter;
```

5.2.20. tx_complete.vhd

```
-- tx_complete.vhd

library ieee;
use ieee.std_logic_1164.all;
```

```
entity tx_complete is
    port ( rst           : in      std_logic;
          setup         : in      std_logic_vector (2 downto 0);
          camwe         : in      std_logic;
          syncck        : in      std_logic;
          cdn           : in      std_logic;
          ctrl          : in      std_logic_vector (13 downto 0);
          ck            : in      std_logic;
          ck40          : in      std_logic;
          dout          : out     std_logic;
          sout          : out     std_logic );
end tx_complete;

architecture arc_tx_complete of tx_complete is

    component tx_data_gen
        port ( data       : in      std_logic_vector (2 downto 0);
              ctrl       : in      std_logic_vector (13 downto 0);
              cdn        : in      std_logic;
              syncck     : in      std_logic;
              ck         : in      std_logic;
              d           : buffer  std_logic_vector (15 downto 0);
              decs       : out     std_logic );
    end component;

    component tx_ds_gen
        port ( d          : in      std_logic_vector (15 downto 0);
              decs       : in      std_logic;
              ck         : in      std_logic;
              ck40       : in      std_logic;
              dout       : out     std_logic;
              sout       : out     std_logic );
    end component;

    signal    decs       : std_logic;
    signal    data       : std_logic_vector (2 downto 0);
    signal    d          : std_logic_vector (15 downto 0);

begin

    inst1 : tx_data_gen
        port map ( data(2 downto 0)=>data(2 downto 0),
                  ctrl(13 downto 0)=>ctrl(13 downto 0),
                  cdn=>cdn, syncck=>syncck, ck=>ck,
                  d(15 downto 0)=>d(15 downto 0),
                  decs=>decs );

    inst2 : tx_ds_gen
        port map ( d(15 downto 0)=>d(15 downto 0), decs=>decs, ck=>ck,
                  ck40=>ck40, dout=>dout, sout=>sout );

    process(rst, camwe)
    begin
        if rst = '0' then
            data <= (others => '0');
        elsif (camwe'event and camwe = '1') then
            data <= setup;
        end if;
    end process;
end arc_tx_complete;
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 115/131
Filename: *ASTEDetailedDesign6*

```
end process;
```

```
end arc_tx_complete;
```

5.2.21. tx_data_gen.vhd

```
-- tx_data_gen.vhd
```

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
entity tx_data_gen is
```

```
    port ( data      : in      std_logic_vector (2 downto 0);  
          ctrl       : in      std_logic_vector (13 downto 0);  
          cdn        : in      std_logic;  
          syncck     : in      std_logic;  
          ck         : in      std_logic;  
          d          : buffer  std_logic_vector (15 downto 0);  
          decs       : out     std_logic );
```

```
end tx_data_gen;
```

```
architecture arc_tx_data_gen of tx_data_gen is
```

```
    component my_xor14
```

```
        port( data      : in std_logic_vector(13 downto 0);  
              result    : out std_logic) ;
```

```
    end component;
```

```
    signal a          : std_logic;
```

```
    signal parity     : std_logic;
```

```
    signal q          : std_logic_vector (13 downto 0);
```

```
begin
```

```
    inst1 : my_xor14 port map ( data => d(15 downto 2), result => a);
```

```
    process (cdn, data, ctrl)
```

```
    begin
```

```
        case (cdn) is
```

```
            when '0' => q<= "00010001000" & data;
```

```
            when '1' => q<= ctrl;
```

```
            when others => q <= (others => '-');
```

```
        end case;
```

```
    end process;
```

```
    process (ck)
```

```
    begin
```

```
        if (ck'event and ck = '0') then
```

```
            d(15 downto 2) <= q;
```

```
            d(0) <= parity;
```

```
            d(1) <= cdn;
```

```
        end if;
```

```
    end process;
```

```
    process (ck)
```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 116/131
Filename: ASTEDetailedDesign6

```
begin
  if (ck'event and ck = '0') then
    decs <= syncck;
  end if;
end process;

parity<= a xnor cdn;

end arc_tx_data_gen;
```

5.2.22. my_xor14.vhd

```
-- Version: 6.1 6.1.1.24

library ieee;
use ieee.std_logic_1164.all;
library a54SXA;

entity my_xor14 is
  port( Data : in std_logic_vector(13 downto 0); Result : out
        std_logic) ;
end my_xor14;

architecture DEF_ARCH of my_xor14 is

  component CM8INV
    port(A : in std_logic; Y : out std_logic) ;
  end component;

  component CM8F
    port(D0, D1, D2, D3, S00, S01, S10, S11 : in std_logic;
        Y, FY : out std_logic) ;
  end component;

  component VCC
    port( Y : out std_logic);
  end component;

  component GND
    port( Y : out std_logic);
  end component;

  signal Data_15_net, Data_16_net, Data_17_net, Data_14_net,
        Data_18_net, Data_19_net, CM8INV_1_Y, CM8INV_5_Y,
        CM8INV_2_Y, CM8INV_3_Y, CM8INV_0_Y, CM8INV_4_Y, VCC_1_net,
        GND_1_net : std_logic ;
  begin

  VCC_2_net : VCC port map(Y => VCC_1_net);
  GND_2_net : GND port map(Y => GND_1_net);
  CM8INV_2 : CM8INV
    port map(A => Data_17_net, Y => CM8INV_2_Y);
  CM8F_Data_16_inst : CM8F
    port map(D0 => CM8INV_1_Y, D1 => Data(8), D2 => Data(8),
        D3 => CM8INV_1_Y, S00 => VCC_1_net, S01 => Data(6),
```

```

S10 => Data(7), S11 => GND_1_net, Y => OPEN , FY =>
Data_16_net);
CM8INV_5 : CM8INV
port map(A => Data(5), Y => CM8INV_5_Y);
CM8F_Data_15_inst : CM8F
port map(D0 => CM8INV_5_Y, D1 => Data(5), D2 => Data(5),
D3 => CM8INV_5_Y, S00 => VCC_1_net, S01 => Data(3),
S10 => Data(4), S11 => GND_1_net, Y => OPEN , FY =>
Data_15_net);
CM8INV_4 : CM8INV
port map(A => Data(11), Y => CM8INV_4_Y);
CM8F_Data_18_inst : CM8F
port map(D0 => CM8INV_3_Y, D1 => Data_14_net, D2 =>
Data_14_net, D3 => CM8INV_3_Y, S00 => VCC_1_net, S01 =>
Data(12), S10 => Data(13), S11 => GND_1_net, Y => OPEN ,
FY => Data_18_net);
CM8F_Data_17_inst : CM8F
port map(D0 => CM8INV_4_Y, D1 => Data(11), D2 => Data(11),
D3 => CM8INV_4_Y, S00 => VCC_1_net, S01 => Data(9),
S10 => Data(10), S11 => GND_1_net, Y => OPEN , FY =>
Data_17_net);
CM8F_Data_14_inst : CM8F
port map(D0 => Data(2), D1 => CM8INV_0_Y, D2 => CM8INV_0_Y,
D3 => Data(2), S00 => VCC_1_net, S01 => Data(0), S10 =>
GND_1_net, S11 => Data(1), Y => OPEN , FY => Data_14_net);
CM8F_Data_19_inst : CM8F
port map(D0 => CM8INV_2_Y, D1 => Data_17_net, D2 =>
Data_17_net, D3 => CM8INV_2_Y, S00 => VCC_1_net, S01 =>
Data_15_net, S10 => Data_16_net, S11 => GND_1_net, Y =>
OPEN , FY => Data_19_net);
CM8INV_0 : CM8INV
port map(A => Data(2), Y => CM8INV_0_Y);
CM8INV_1 : CM8INV
port map(A => Data(8), Y => CM8INV_1_Y);
CM8INV_3 : CM8INV
port map(A => Data_14_net, Y => CM8INV_3_Y);
CM8F_Result : CM8F
port map(D0 => VCC_1_net, D1 => GND_1_net, D2 => GND_1_net,
D3 => VCC_1_net, S00 => VCC_1_net, S01 => Data_18_net,
S10 => Data_19_net, S11 => GND_1_net, Y => OPEN , FY =>
Result);
end DEF_ARCH;

```

5.2.23. tx_ds_gen.vhd

```

-- tx_ds_gen.vhd

library ieee;
use ieee.std_logic_1164.all;

entity tx_ds_gen is
port ( d          : in          std_logic_vector (15 downto 0);
      decs       : in          std_logic;
      ck        : in          std_logic;
      ck40      : in          std_logic;
      dout      : out         std_logic;

```

```

        sout          : out      std_logic );
end tx_ds_gen;

architecture arc_tx_ds_gen of tx_ds_gen is

    component clkint
        port ( a          : in      std_logic;
              y          : out     std_logic );
    end component;

    signal    a1          : std_logic;
    signal    a2          : std_logic;
    signal    a3          : std_logic;
    signal    a3q         : std_logic;
    signal    q0          : std_logic;
    signal    q1          : std_logic;
    signal    q2          : std_logic;
    signal    q3          : std_logic;
    signal    q           : std_logic_vector(15 downto 0);
    signal    r           : std_logic_vector(13 downto 0);
    signal    ckint       : std_logic;

begin

    inst0: clkint port map (a=>ck40, y=>ckint);

    process (ckint)
    begin
        if (ckint'event and ckint = '0') then
            q1 <= not ck;
            q2 <= not q1;
        end if;
    end process;

    process (ckint, a1)
    begin
        if (ckint'event and ckint = '1') then
            if (a1 = '1') then
                q <= d;
            else
                q <= '0' & q(15 downto 1);
            end if;
        end if;
    end process;

    process (ckint)
    begin
        if (ckint'event and ckint = '1') then
            r <= decs & r(13 downto 1);
        end if;
    end process;

    process (ckint)
    begin
        if (ckint'event and ckint = '1') then
            q0 <= q(0);
            a3q <= a3;
        end if;
    end process;
end arc_tx_ds_gen;

```



AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 119/131
Filename: *ASTEDetailedDesign6*

```
    q3 <= a2;
  end if;
end process;

a1 <= q1 and q2;
a2 <= ((q(0) xor q0) xnor q3) and (not a3q);
a3 <= r(0) and decs;
dout <= q0;
sout <= q3;

end arc_tx_ds_gen;
```



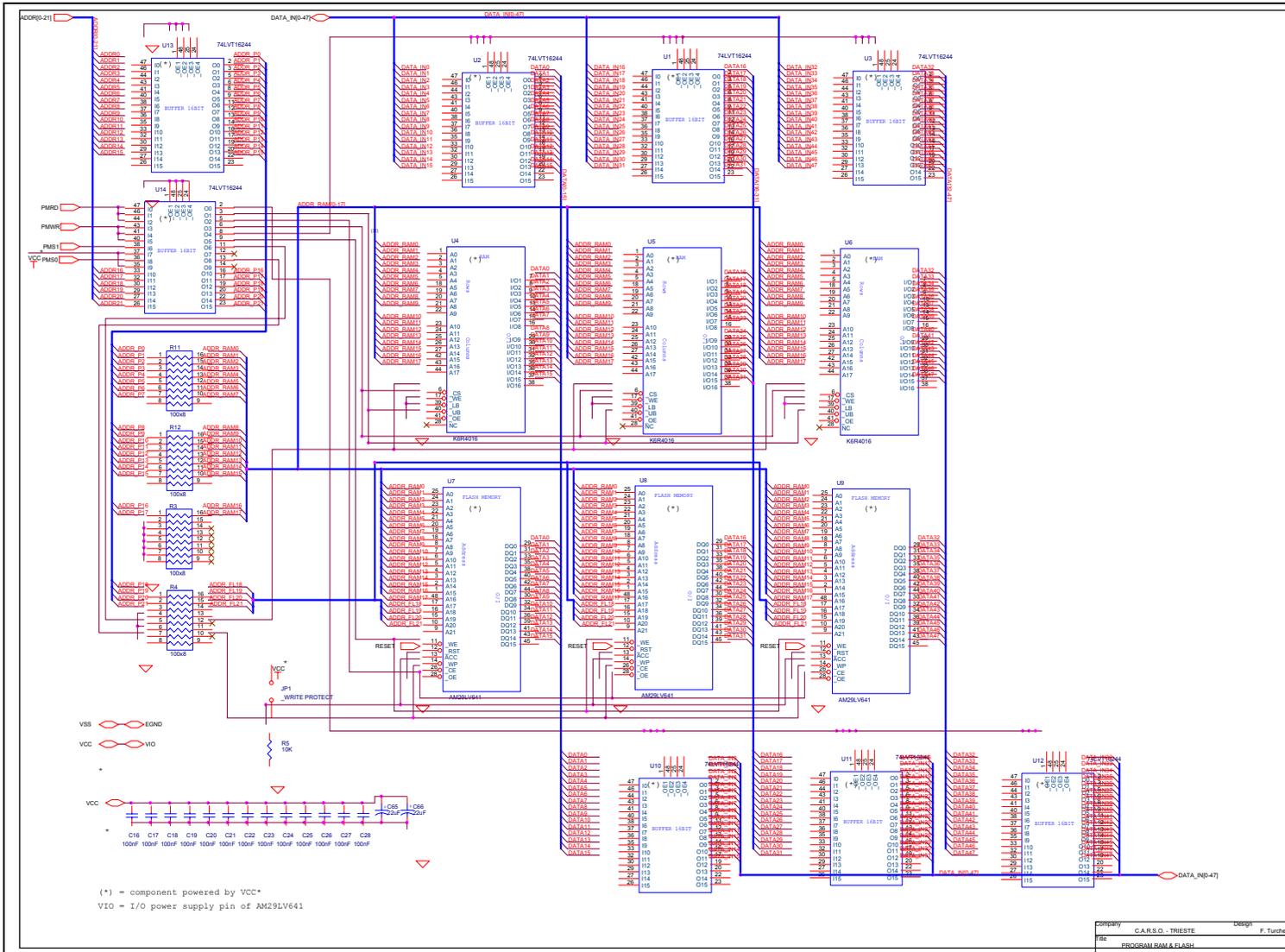
AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 6
DATE: Oct 06
PAGE: 120/131
Filename: *ASTEDetailedDesign6*

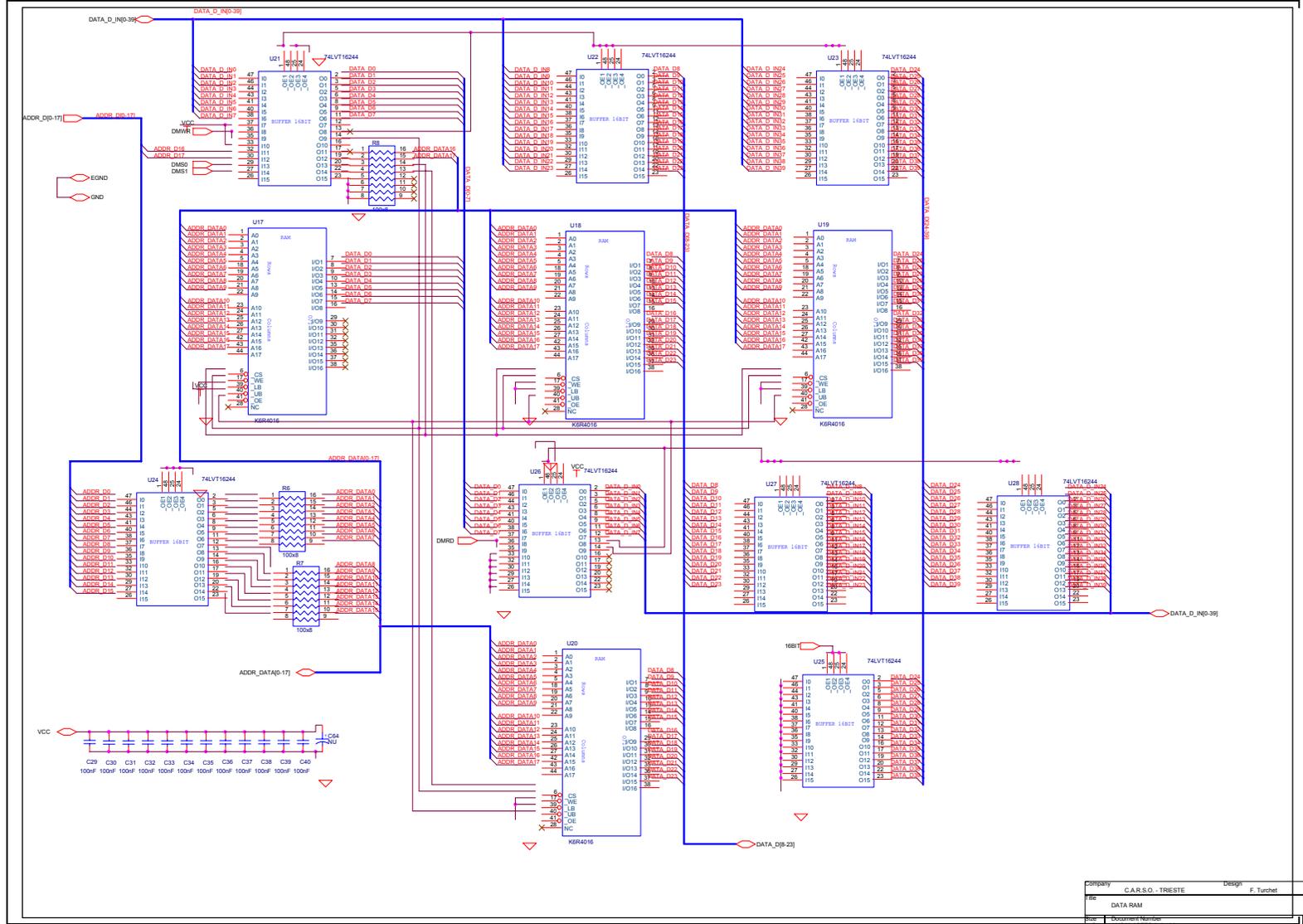
6. SCHEMATICS

AMICA FOR AMS ASTE Detailed Design

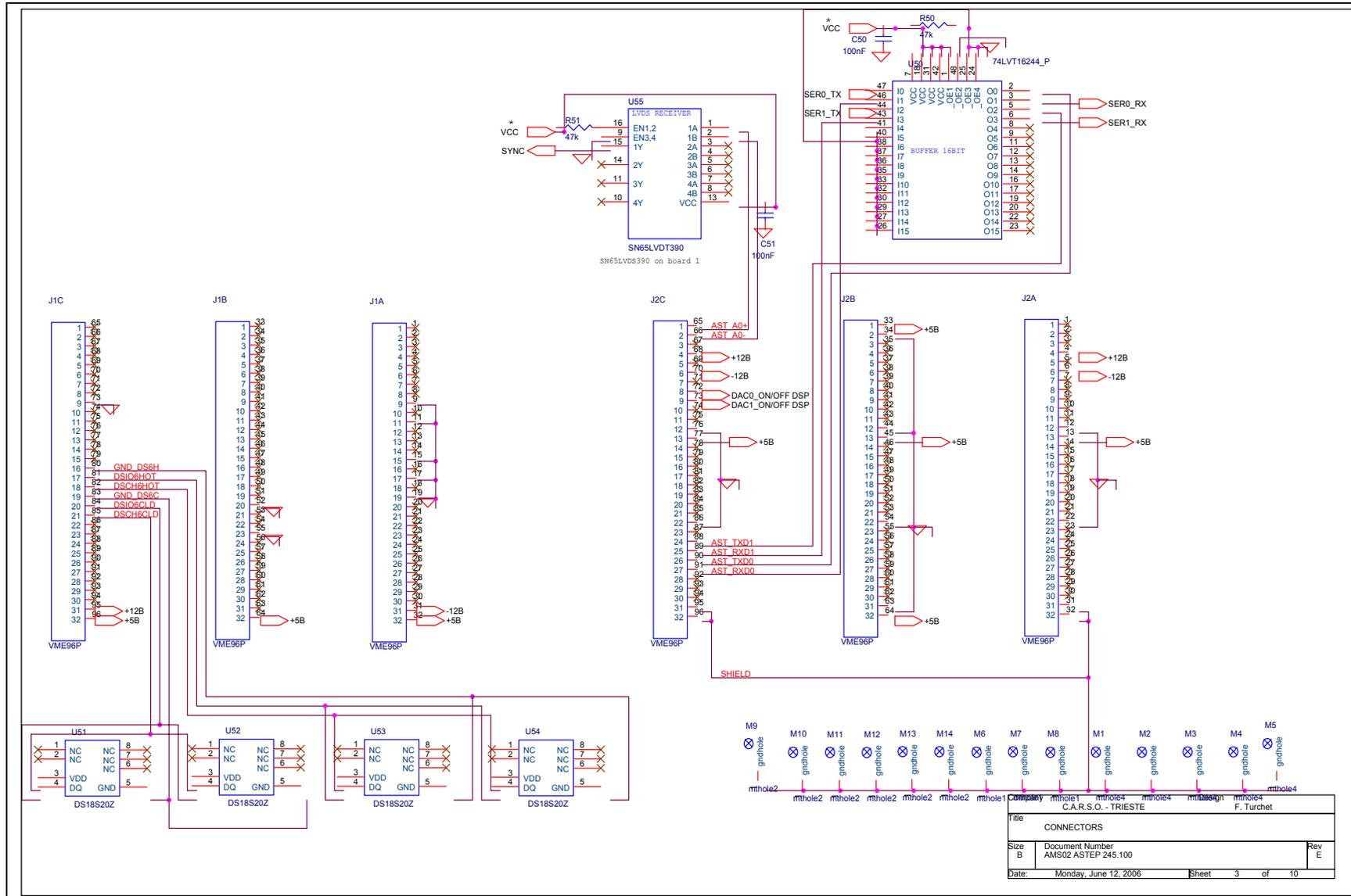
DOC.: AMST/STED /1/A
ISSUE: 5
DATE: 18/06/07
PAGE: 121/131
Filename: ASTEDetailedDesign6



Company	CARSO - TRIESTE	Design	F. Turcher
File	PROGRAM RAM & FLASH		
Doc	Documenti Technici		
C	AMIS2 ASTEP 246.100		
Rev	Monday, June 12, 2006	Sheet	1 of 10



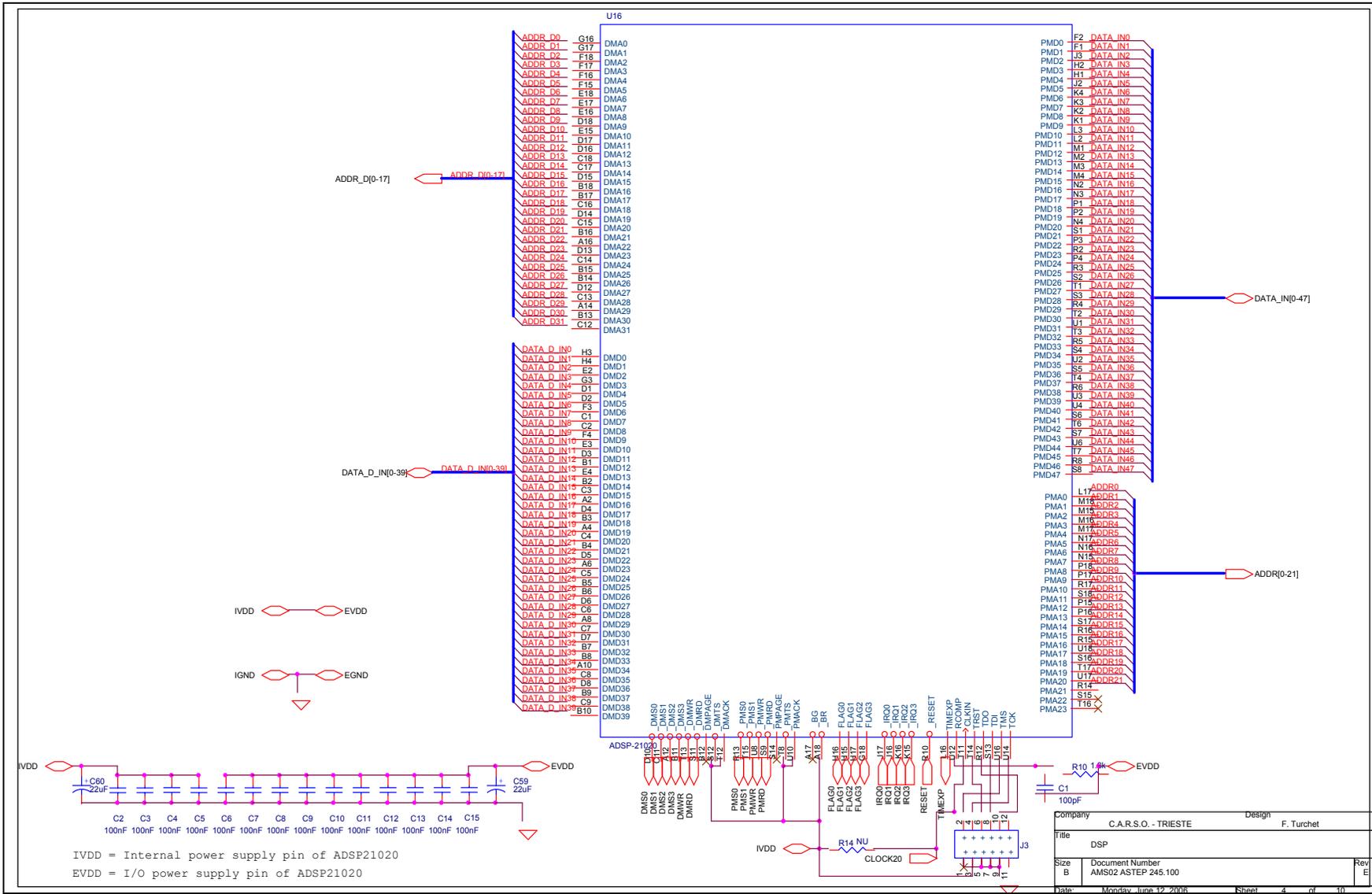
Company	C.A.R.S.O. - TRIESTE	Design	F. Turchet
File	DATA RAM		
Sheet	122		131
C	Microelectronics		
U	74LV16244		
Date	Monday, 18 Jun 2007	Sheet	1 of 13



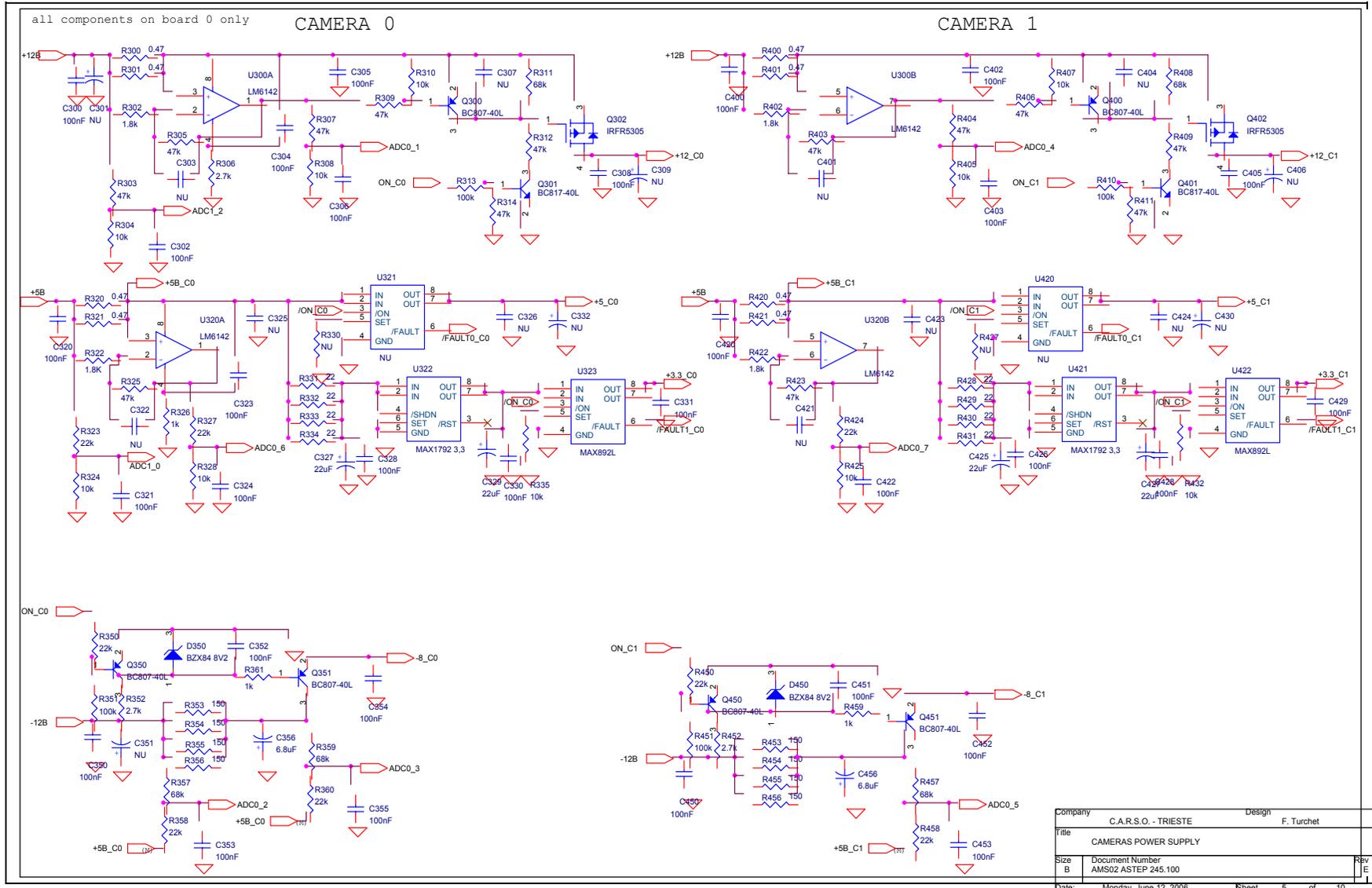


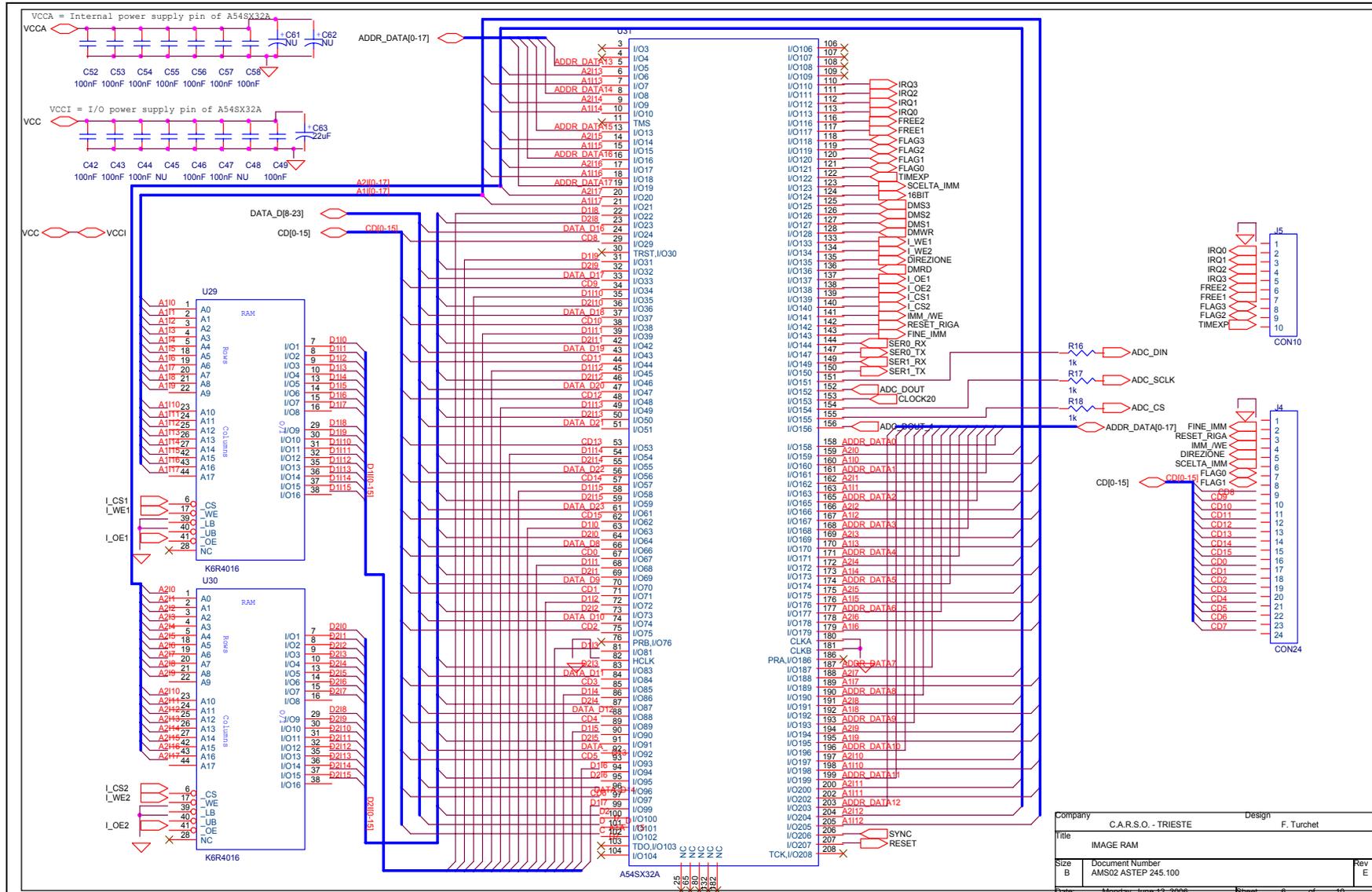
AMICA FOR AMS ASTE Detailed Design

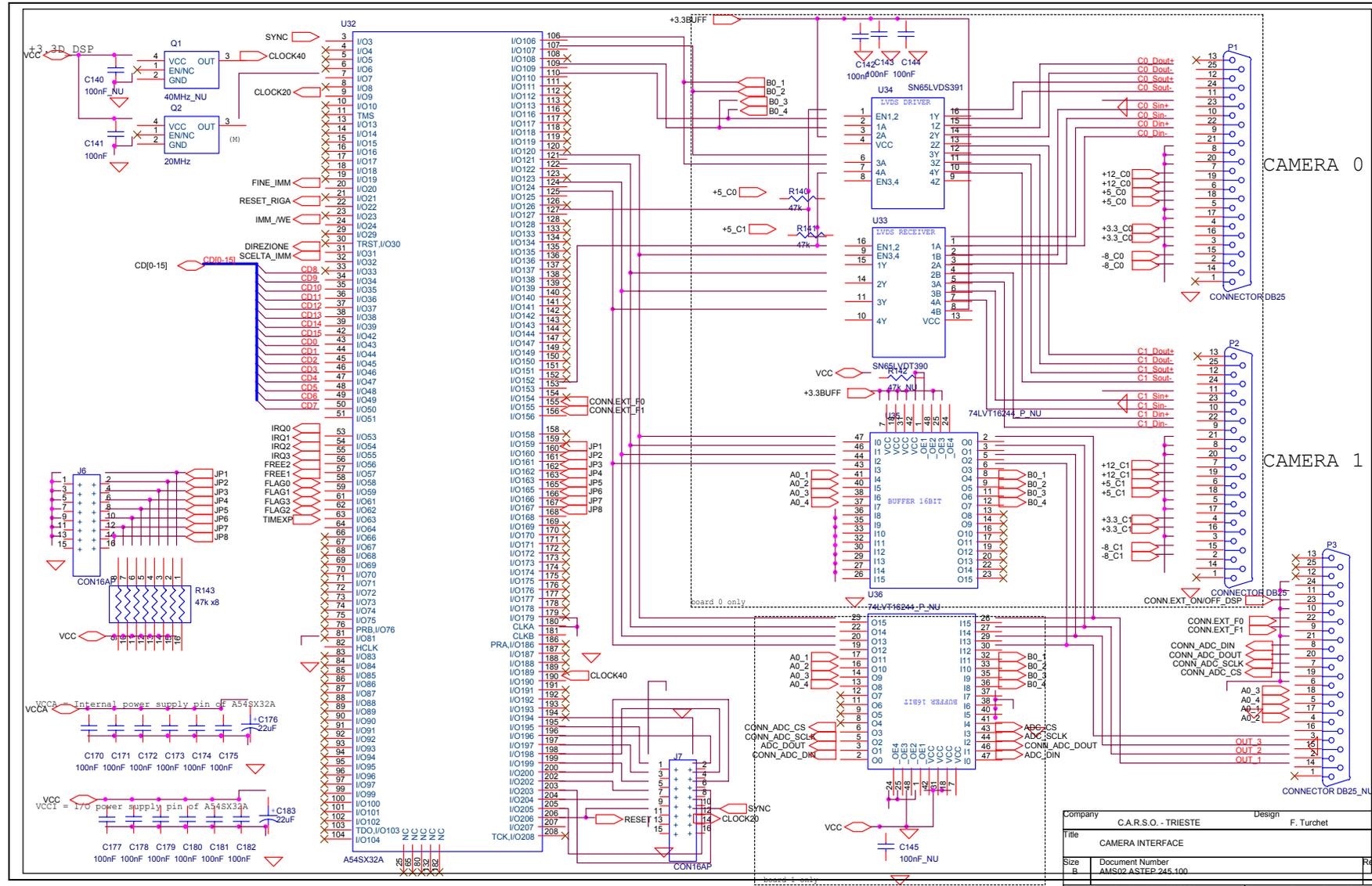
DOC.: AMST/ STED /1/A
 ISSUE: 5
 DATE: 18/06/07
 PAGE: 124/131
 Filename: *ASTEDetailedDesign6*



Any information contained in this document is property of CARSO and is strictly private and confidential. All rights reserved.







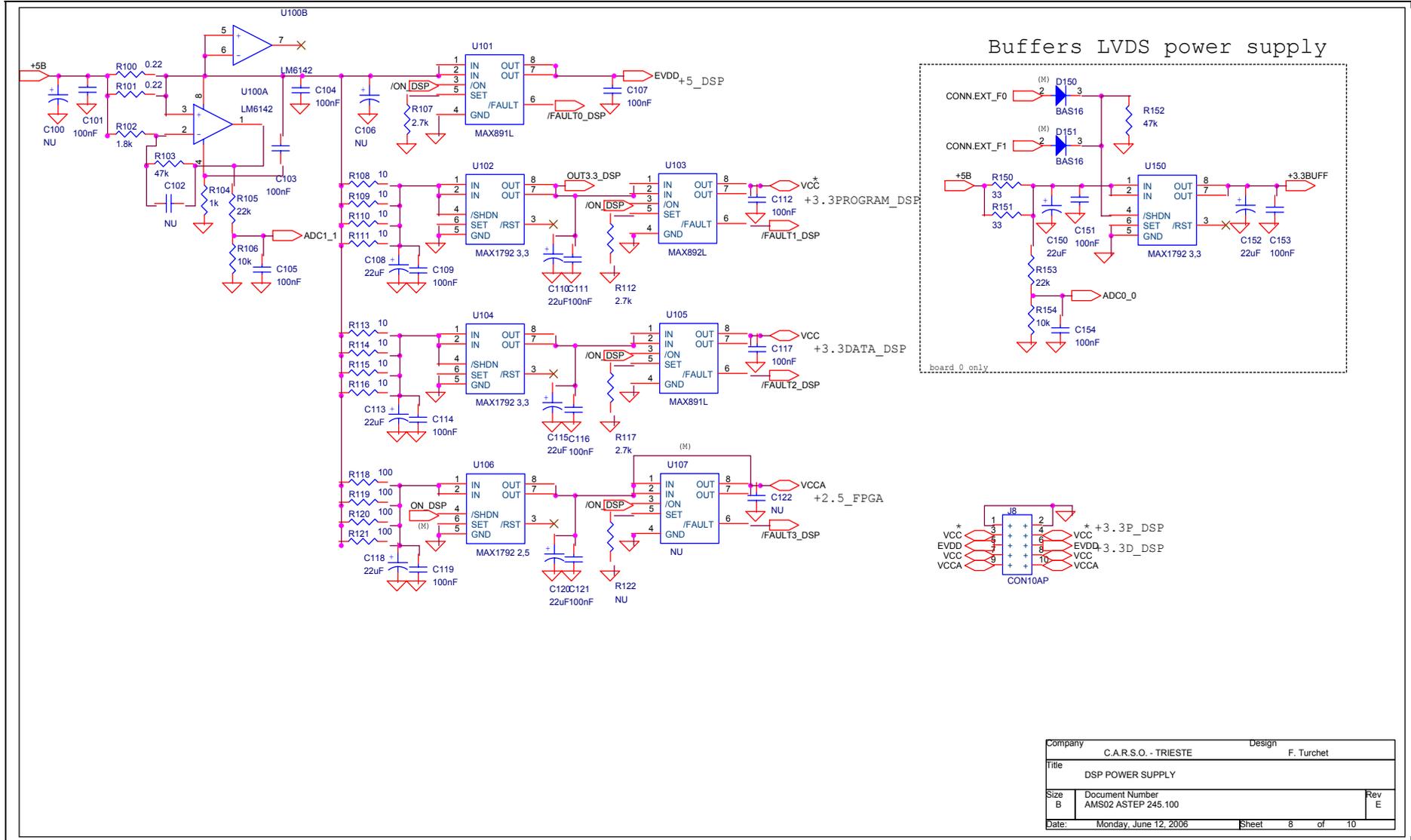
company	C.A.R.S.O. - TRIESTE	Design	F. Turchet
Title	CAMERA INTERFACE		
Size	Document Number	AMS02 ASTEP 245.100	R4v
B			F
Date	Monday, June 12, 2006	Sheet	7 of 10

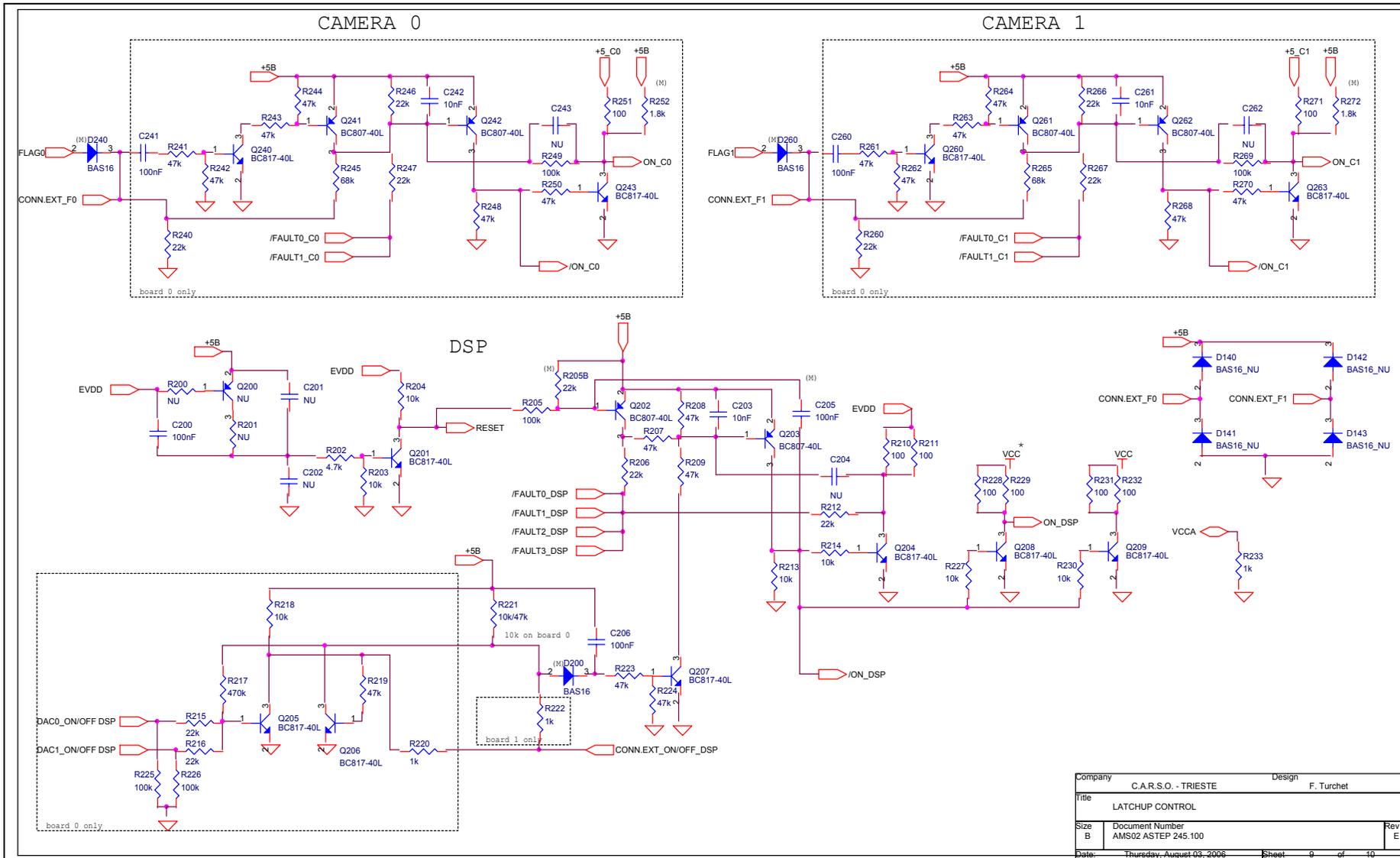
Any information contained in this document is property of CARSO and is strictly private and confidential. All rights reserved.



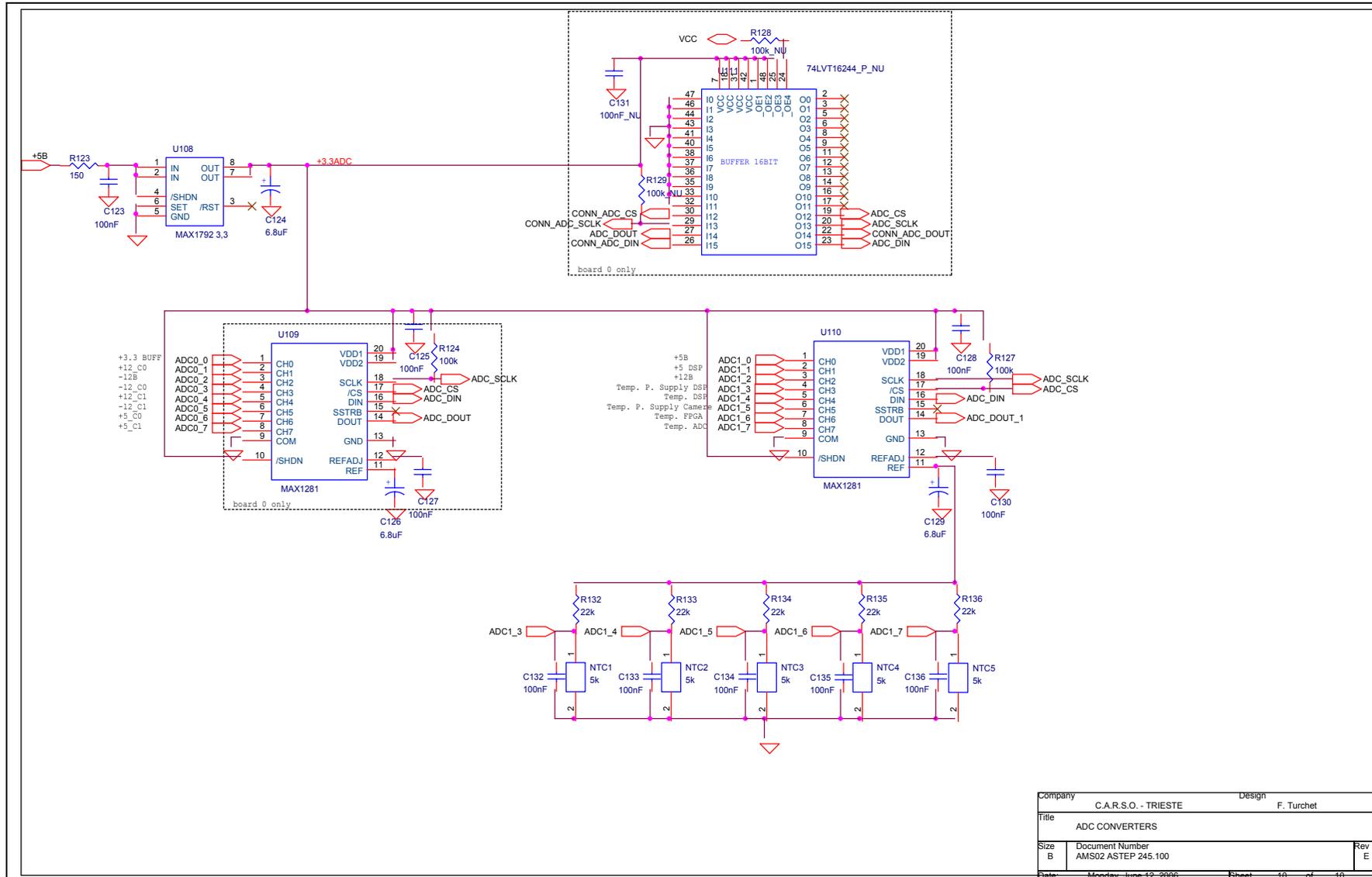
AMICA FOR AMS ASTE Detailed Design

DOC.: AMST/ STED /1/A
ISSUE: 5
DATE: 18/06/07
PAGE: 128/131
Filename: *ASTEDetailedDesign6*





Company		C.A.R.S.O. - TRIESTE		Design		F. Turchet	
Title							
LATCHUP CONTROL							
Size	Document Number					Rev	
B	AMS02 ASTEP 245.100					E	
Date:	Thursday, August 03, 2006	Sheet	9	of	10		



Company	C.A.R.S.O. - TRIESTE	Design	F. Turchet
Title	ADC CONVERTERS		
Size	Document Number		Rev
B	AMS02 ASTEP 245.100		E
Date:	Monday, June 12, 2006	Sheet	10 of 10